

b2b slots INTEGRATION

Table of contents

Table of contents	2
1. Protocol	3
2. Link creation	3
3. Integration APIs.....	4
3.1 Game URL	5
3.2 Auth API.....	6
3.3 Debit API.....	8
3.4 Credit API.....	10
3.6 Get Features API	13
3.7 Activate Features API.....	17
3.8 Update Features API.....	20
3.9 End Features API.....	23
4 Dictionaries	26
4.1 Currency Dictionary.....	26
4.2 Game Codes	27
4.2 Error Codes	28
4.3 List of supported languages.....	29
5. API Examples.....	30
5.1 Auth	30
5.2 Debit.....	31
5.3 Credit.....	32
5.4 Get Features.....	33
5.5 Activate Features	34
5.6 Update Features.....	35
5.7 End Features	36

1. Protocol

Integration uses Http POST communications based on the Json message type. The operator can only use HTTPS to communicate with our servers.

For any synchronous or asynchronous communication regarding financial transactions, such as debit and credit, the integrator backend should receive a response with the same transaction_id parameter that was sent in the request.

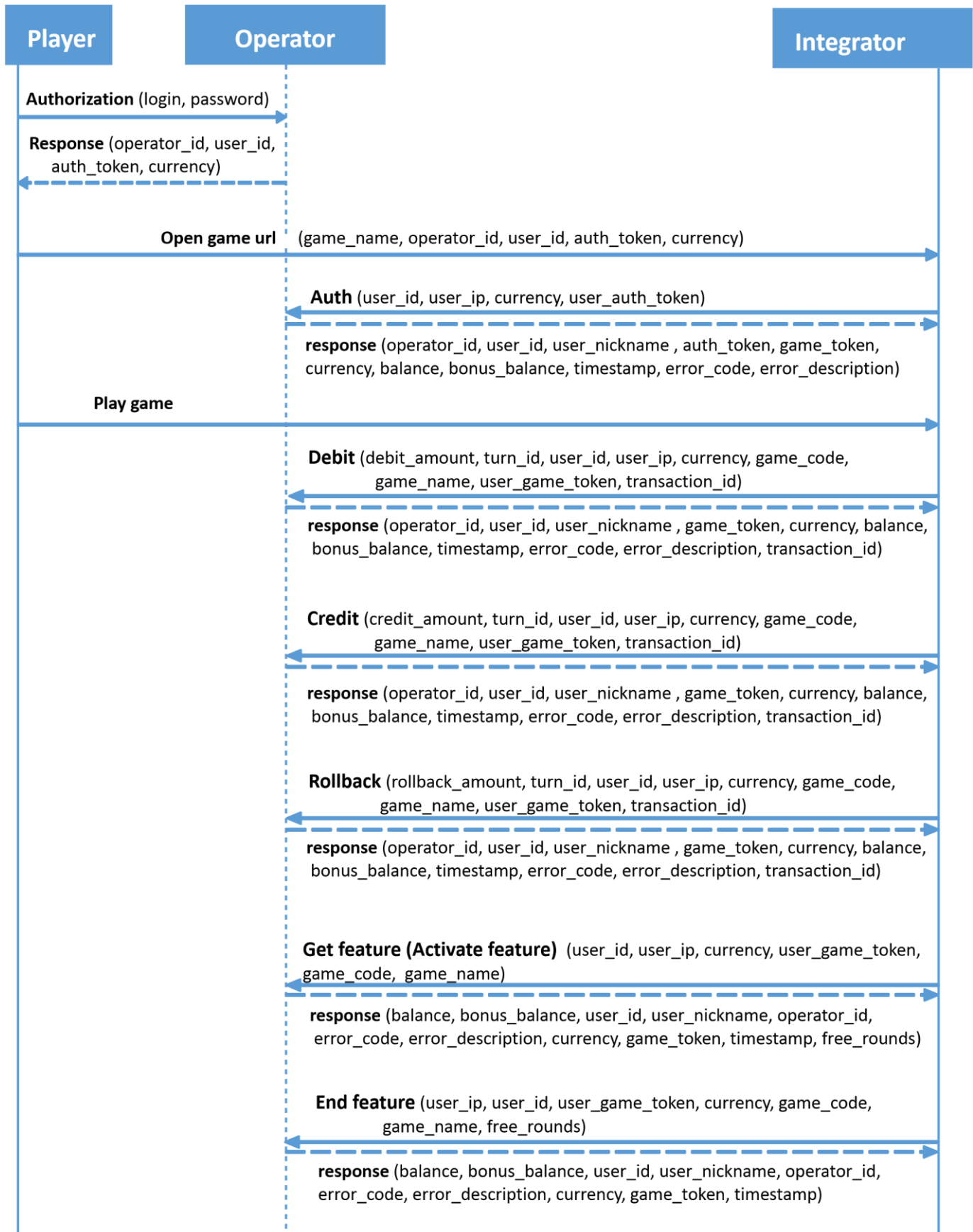
The Json format is used to communicate with the backend for all kinds of APIs.

2. Link creation

To create backend calls, you need to get a URL, the type of connection (protocol name) and the exact names of the final operations. The following is a list of possible internal operations:

- auth
- debit
- credit
- get features
- activate features
- update features
- end features

3. Integration APIs



- Players make authorization on the operator's side and must receive the following parameters for use in the game's URL:

Имя параметра	Тип данных/ Описание
operator_id	int (11)
	The number of the operator to which the player belongs.
user_id	string (128)
	Unique identifier for each player on the operator side
auth_token	string (255)
	A unique identifier that is generated and sent by the operator, to define an interaction session
currency	string
	Player balance currency code in a specific session

3.1 Game URL

The following table describes the options provided by the operator for the Integrator to start the game.

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

Имя параметра	Тип данных/ Описание
game_name	string
	The unique name of the game the player wants to run.
operator_id	int (11)
	The number of the operator to which the player belongs.
user_id	string (128)
	Unique identifier for each player on the operator side
auth_token	string (255)
	A unique identifier that is generated and sent by the operator to determine the interaction session.
currency	string
	Player balance currency code in a specific session BTC, LTC, USD, EUR, RUB, KZT, UAH (may be expanded in the future)
lang	string
	Interface language, ru, en (optional parameter)
home_url	string
	URL address to which you will be redirected by pressing the Home button in the mobile version of the games

Example:

```
https://int.apiforb2b.com/games/BookOfRaClassic.game?operator_id=0&user_id=1&auth_token=2a9a36b3487129aa899a57
b161325ac6e3772174&currency=BTC&lang=en&home_url=http://mysite.com
```

3.2 Auth API

- **Description**

The Auth API allows the Integrator system to verify information obtained from the game URL. Only after the operator approves this information, the game will open for the player.

Parameter name		Data Type / Description
Input parameters		
api		String, value = "do-auth-user-ingame"
		unique name api
data		JsonObject
		Json object of the following parameters
	user_id	string (128)
		Unique identifier for each player on the operator side
	user_ip	string (30)
		Internet IP computer end player
	user_auth_token	string (255)
		Unique identifier generated and sent by the operator to identify the session
	currency	string
		Player balance currency code in a specific session: BTC, LTC, USD, EUR, RUB, KZT, UAH (may be expanded in the future)
	game_code	Int
		Unique Integrator Game Code
	game_name	String
		Unique name of the game
Output parameters		
api		String
		the unique name sent to the Integrator in the request must have the value "do-auth-user-ingame"
success		Boolean
		As a result of api request, the value should always be true
answer		JsonObject the following parameters
	operator_id	Int (11)
		Unique identifier of the operator
	user_id	String (128)
		Unique identifier for each player on the operator side
	user_nickname	String (32)
		Player nickname. Letters and numbers only
	balance	String (20)
		Available player balance at the time of the request. The value should be presented in the currency of the current session. String representation BigDecimal(15,5)
	bonus_balance	String (20)
		Доступный бонусный баланс игрока на момент запроса. Значение должно быть представлено в валюте текущей сессии. Строковое представление BigDecimal(15,5)
	auth_token	String (255)
		Initial user user_auth_token from the operator (by launching the game URL) - we use this for verification.
	game_token	String (255)
		Не путать с вводом auth_token. Этот токен является токеном сеанса. Уникальный идентификатор, который генерируется оператором для игровой сессии.
	error_code	int
		Determines whether the request was processed successfully. If no error is detected, the code value should be 0
	error_description	String (255)
		A string describing the answer. This line is not a message to the player, but contains detailed information about the error.
	currency	String
		Balance currency
	timestamp	String (15)
		String representation of time in UNIX milliseconds.

3.3 Debit API

- **Description**

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

Parameter name		Data Type / Description
Input parameters		
api		String, value = "do-debit-user-ingame"
		unique name api
data		JsonObject
		Json object of the following parameters
	user_id	string (128)
		Unique identifier for each player on the operator side
	user_ip	string (30)
		Internet IP computer end player
	user_game_token	string (255)
		The unique identifier that is generated (in auth api) by the operator to determine the game session.
	currency	string
		Player balance currency code in a specific session: BTC, LTC, USD, EUR, RUB, KZT, UAH (may be expanded in the future)
	turn_id	BigInt
		A unique key to identify the game cycle.
	transaction_id	String (36)
		A unique key for a specific financial transaction. This key ensures that the transaction will be processed only once.
	game_code	Int
		Unique Integrator Game Code
	game_name	String
		Unique name of the game
	debit_amount	String (20)
		Amount requested to withdraw from a player's wallet. The value must be a positive number above zero. String representation BigDecimal(15,5)
	debit_type	String
		Type of debit. It can take the values "regular" for regular spins, and "freeRounds" for freerounds. It can also be set to getBalance. In this case, debit_amount = 0 and round_id = 0 - this request is used to get the balance from the operator, without changing it.
	round_id	BigInt
		A unique key to identify the game round.
Output parameters		
api		String
		The unique name api sent in the request to the Integrator, the value must be set to "do-debit-user-ingame"
success		Boolean
		В результате запроса api значение всегда должно быть true
answer		JsonObject the following parameters
	operator_id	Int (11)
		the following parameters
	transaction_id	String (36)
		The same transaction_id that was sent in the request. A unique key to indicate a specific financial transaction. This key ensures that the transaction will be processed only once.
	user_id	String (128)
		Unique identifier for each player on the operator side
	user_nickname	String (32)
		Player nickname. Letters and numbers only
	balance	String (20)
		Available player balance at the time of the request. The value should be presented in the currency of the current session. String representation BigDecimal(15,5)
		String (20)

	bonus_balance	Available bonus player balance at the time of request. The value should be presented in the currency of the current session. String representation BigDecimal(15,5)
	bonus_amount	String (20)
		Bonus amount is part of the total debit amount. * Cannot be more than the debit amount. String representation BigDecimal(15,5)
	game_token	String (255)
		Not to be confused with typing auth_token. This token is the session token. The unique identifier that is generated by the operator for the game session.
	error_code	int
		Determines whether the request was processed successfully. If no error is detected, the value must be 0
	error_description	String (255)
		A string describing the answer. This line is not a message to the player, but contains detailed information about the error.
	currency	String
		Balance currency
	timestamp	String (15)
		String representation of time in UNIX milliseconds.

3.4 Credit API

- **Description**

The Credit API allows the Integrator platform to carry out a financial operation to return money to the player's wallet through the operator's server. When calculating the results, the player who loses has a credit amount equal to 0. The winning player receives the payout amount.

Parameter name		Data Type / Description
Input parameters		
api		String, value = "do-credit-user-ingame"
		unique name api
data		JsonObject
		Json object of the following parameters
	user_id	string (128)
		Unique identifier for each player on the operator side
	user_ip	string (30)
		Internet IP computer end player
	user_game_token	string (255)
		The unique identifier that is generated (in auth api) by the operator to determine the game session.
	currency	string
		Player balance currency code in a specific session: BTC, LTC, USD, EUR, RUB, KZT, UAH (may be expanded in the future)
	turn_id	BigInt
		A unique key to identify the game cycle.
	transaction_id	String (36)
		A unique key for a specific financial transaction. This key ensures that the transaction will be processed only once.
	game_code	Int
		Unique Integrator Game Code
	game_name	String
		Unique name of the game
	credit_amount	String(20)
		Amount requested to replenish a player's wallet. The value must be a positive number above zero. String representation BigDecimal(15,5)
	credit_type	String
		Type of loan. It can take the values "regular" for regular spins, and "freeRounds" - to win in freerounds
	round_id	BigInt
		A unique key to identify the game round.
Выходные параметры		
api		String
		unique name api sent to the Integrator, the value must be set to "do-credit-user-ingame"
success		Boolean
		As a result of api request, the value should always be true
answer		JsonObject the following parameters
	operator_id	Int (11)
		Unique identifier of the operator
	transaction_id	String (36)
		The same transaction_id that was sent in the request. A unique key to indicate a specific financial transaction. This key ensures that the transaction will be processed only once.
	user_id	String (128)
		Unique identifier for each player on the operator side
	user_nickname	String (32)
		Player nickname. Letters and numbers only
	balance	String (20)
		Available player balance at the time of the request. The value should be presented in the currency of the current session. String representation BigDecimal(15,5)
	bonus_balance	String (20)
		Available bonus balance in the player's wallet at the time of request. The value will be presented in the currency of the current session. String representation BigDecimal(15,5)

	bonus_amount	String(20)
		Bonus amount of the total credit_amount. * Cannot be greater than credit_amount. String representation BigDecimal(15,5)
	game_token	String (255)
		Not to be confused with typing auth_token. This token is the session token. The unique identifier that is generated by the operator for the game session.
	error_code	int
		Determines whether the request was processed successfully. If no error is detected, this code value is 0
	error_description	String (255)
		A string describing the answer. This line is not a message to the player, but contains detailed information about the error.
	currency	String
		Balance currency
	timestamp	String (15)
		String representation of time in UNIX milliseconds.

3.6 Get Features API

- **Описание**

The Get Features API allows the Integrator platform to request any bonus program. Freerounds are currently supported (free spins). There are two implementations of freerounds, version 0 and version 2.

- **Work algorithm version 0.**

1) When the game starts, «do-get-features-user-ingame» is called if there are no active freerounds already. In case there are already active freerounds, they will be played out.

2) A message is displayed to the player stating that he has a bonus - "free-rounds"

3) If the player activates the bonus, «do-activate-features-user-ingame» is called and bonus spins begin.

4) At the end of the bonus comes "do-end-features-user-ingame" and "do-credit-user-ingame" with the total amount of winnings in freerounds.

5) If in paragraph 3 the player did not activate the bonus, then the usual game continues.

- **Work algorithm version 2.**

1) When the game starts, «do-get-features-user-ingame» is called. Unlike version 0, in version 2, do-get-features-user-ingame is called every time you enter the game. The following options are possible here:

a) if there were no active freerounds, then the behavior is similar to version 0.

b) if there were already active freerounds, but there are no freerounds in the answer, then the active freerounds are canceled and the usual game starts.

c) if there have already been active freerounds, and in the response freerounds with a different ID, then the active freerounds are canceled and the player is prompted to start new ones.

d) if there were already active freerounds, and in the response of freerounds with the same ID, then active freerounds are played out.

2) A message is displayed to the player stating that he has a bonus - "free-rounds"

3) If the player activates the bonus, «do-activate-features-user-ingame» is called and bonus spins begin.

4) For each spin, «do-update-features-user-ingame» is called, which reports the current state of the freerounds, the total number of rounds, the number of rounds played, the total win and win in the current round.

5) At the end of the bonus comes «do-end-features-user-ingame» and "do-credit-user-ingame" with the total amount of winnings in freerounds.

6) If in paragraph 3 the player did not activate the bonus, then the usual game continues.

Parameter name		Data Type / Description
Input parameters		
api		String, value = "do-get-features-user-ingame"
		unique name api
data		JsonObject
		Json object of the following parameters
user_id		string (128)
		Unique identifier for each player on the operator side
user_ip		string (30)
		Internet IP computer end player
user_game_token		string (255)
		The unique identifier that is generated (in auth api) by the operator to determine the game session.
currency		string
		Player balance currency code in a specific session: BTC, LTC, USD, EUR, RUB, KZT, UAH (may be expanded in the future)
game_code		Int
		Unique Integrator Game Code
Output parameters		
api		String
		unique name api sent to the Integrator, the value must be set to "do-get-features-user-ingame"
success		Boolean
		As a result of api request, the value should always be true
answer		JsonObject the following parameters
operator_id		Int (11)
		Unique identifier of the operator
user_id		String (128)
		Unique identifier for each player on the operator side
user_nickname		String (32)
		Player nickname. Letters and numbers only
balance		String (20)
		Available player balance at the time of the request. The value should be presented in the currency of the current session. String representation BigDecimal(15,5)
bonus_balance		String (20)
		Available bonus balance in the player's wallet at the time of request. The value will be presented in the currency of the current session. String representation BigDecimal(15,5)
game_token		String (255)
		Не путать с вводом auth_token. Этот токен является токеном сеанса. Уникальный идентификатор, который генерируется оператором для игровой сессии.
error_code		int
		Determines whether the request was processed successfully. If no error is detected, this code value is 0
error_description		String (255)
		A string describing the answer. This line is not a message to the player, but contains detailed information about the error.
currency		String
		Balance currency
timestamp		String (15)
		String representation of time in UNIX milliseconds.
free_rounds		JsonObject
		Json Object the following parameters
id		int
		Unique Bonus ID

		count	int
			Number of free spins
		bet	int
			The rate at which the scrolls will be launched
		lines	int
			The number of lines on which scrolls will be launched
		mpl	int
			Bonus Multiplier
		cp	double
			Denomination
		version	int
			Freeround version. It can take values 0 and 2. Differences in the operation of versions are indicated in the API description

3.7 Activate Features API

- **Description**

Activate Features API allows the Integrator platform to notify the operator of bonuses activated by the player.

Parameter name		Data Type / Description
Input parameters		
api		String, value = "do-activate-features-user-ingame"
		The unique api name
data		JsonObject
		Json object of the following parameters
user_id		string (128)
		Unique identifier for each player on the operator side
user_ip		string (30)
		Internet IP computer end player
user_game_token		string (255)
		The unique identifier that is generated (in auth api) by the operator to determine the game session.
currency		string
		Player balance currency code in a specific session: BTC, LTC, USD, EUR, RUB, KZT, UAH (may be expanded in the future)
game_code		Int
		A unique game code in the Integrator
free_rounds		JsonObject
		Json Object the following parameters
id		int
		Unique Bonus ID
Output parameters		
api		String
		The unique name api sent in the request to the Integrator, the value must be set to "do-activate-features-user-ingame"
success		Boolean
		As a result of api request, the value should always be true
answer		JsonObject the following parameters
	operator_id	Int (11) Unique identifier of the operator
user_id		String (128)
		Unique identifier for each player on the operator side
user_nickname		String (32)
		Player nickname. Letters and numbers only
balance		String (20)
		Available player balance at the time of the request. The value should be presented in the currency of the current session. String representation BigDecimal(15,5)
bonus_balance		String (20)
		Available bonus balance in the player's wallet at the time of request. The value will be presented in the currency of the current session. String representation BigDecimal(15,5)
game_token		String (255)
		Not to be confused with typing auth_token. This token is the session token. The unique identifier that is generated by the operator for the game session.
error_code		int
		Determines whether the request was processed successfully. If no error is detected, the value must be 0
error_description		String (255)
		A string describing the answer. This line is not a message to the player, but contains detailed information about the error.
currency		String
		Balance currency

	timestamp	String (15)
		String representation of time in UNIX milliseconds.

3.8 Update Features API

- **Description**

Update Features API allows the Integrator platform to notify about the course of the current activated bonus. This API is called after each spin and transmits the current state of freerounds - the number of rounds played out of the total, the total win and win in the current round. This API is called only for freerounds version 2.0 and later.

Parameter name		Data Type / Description
Input parameters		
api		String, value = "do-update-features-user-ingame"
		The unique api name
data		JsonObject
		Json object of the following parameters
user_id		string (128)
		Unique identifier for each player on the operator side
user_ip		string (30)
		Internet IP computer end player
user_game_token		string (255)
		The unique identifier that is generated (in auth api) by the operator to determine the game session.
currency		string
		Player balance currency code in a specific session: BTC, LTC, USD, EUR, RUB, KZT, UAH (may be expanded in the future)
game_code		Int
		A unique game code in the Integrator
free_rounds		JsonObject
		Json Object нижеследующих параметров
id		int
		Unique Bonus ID
win		double
		The amount of winnings in the bonus. Amount to replenish the player's wallet. The value must be a positive number above or equal to zero
count		int
		Total free spins
played		int
		Number of free spins played
round_win		double
		Amount of winnings in the current round.
round_id		BigInt
		A unique key to identify the game round.
Выходные параметры		
api		String
		The unique name api sent in the request to the Integrator, the value must be set to "do-end-features-user-ingame"
success		Boolean
		As a result of api request, the value should always be true
answer		JsonObject the following parameters
	operator_id	Int (11) Unique identifier of the operator
user_id		String (128)
		Unique identifier for each player on the operator side
user_nickname		String (32)
		Player nickname. Letters and numbers only
balance		String (20)
		Available player balance at the time of the request. The value should be presented in the currency of the current session. String representation BigDecimal(15,5)
		String (20)

	bonus_balance	Available bonus balance in the player's wallet at the time of request. The value will be presented in the currency of the current session. String representation BigDecimal(15,5)
	game_token	String (255) Not to be confused with typing auth_token. This token is the session token. The unique identifier that is generated by the operator for the game session.
	error_code	int Determines whether the request was processed successfully. If no error is detected, the value must be 0
	error_description	String (255) A string describing the answer. This line is not a message to the player, but contains detailed information about the error.
	currency	String Balance currency
	timestamp	String (15) String representation of time in UNIX milliseconds.

3.9 End Features API

- **Description**

End Features API allows the Integrator platform to notify about the end of previous activated bonuses.

Parameter name		Data Type / Description
Input parameters		
api		String, value = "do-end-features-user-ingame"
		The unique api name
data		JsonObject
		Json object of the following parameters
user_id		string (128)
		Unique identifier for each player on the operator side
user_ip		string (30)
		Internet IP computer end player
user_game_token		string (255)
		The unique identifier that is generated (in auth api) by the operator to determine the game session.
currency		string
		Player balance currency code in a specific session: BTC, LTC, USD, EUR, RUB, KZT, UAH (may be expanded in the future)
game_code		Int
		A unique game code in the Integrator
free_rounds		JsonObject
		Json Object the following parameters
id		int
		Уникальный идентификатор бонуса
win		double
		The amount of winnings in the bonus. Amount to replenish the player's wallet. The value must be a positive number above or equal to zero
Выходные параметры		
api		String
		The unique name api sent in the request to the Integrator, the value must be set to "do-end-features-user-ingame"
success		Boolean
		As a result of api request, the value should always be true
answer		JsonObject the following parameters
operator_id		Int (11)
		Unique identifier of the operator
user_id		String (128)
		Unique identifier for each player on the operator side
user_nickname		String (32)
		Player nickname. Letters and numbers only
balance		String (20)
		Доступный баланс игрока на момент запроса. Значение должно быть представлено в валюте текущей сессии. Строковое представление BigDecimal(15,5)
bonus_balance		String (20)
		Available bonus balance in the player's wallet at the time of request. The value will be presented in the currency of the current session. String representation BigDecimal(15,5)
game_token		String (255)
		Not to be confused with typing auth_token. This token is the session token. The unique identifier that is generated by the operator for the game session.
error_code		int
		Determines whether the request was processed successfully. If no error is detected, the value must be 0
error_description		String (255)
		A string describing the answer. This line is not a message to the player, but contains detailed information about the error.

	currency	String
		Balance currency
	timestamp	String (15)
		String representation of time in UNIX milliseconds.

4 Dictionaries

4.1 Currency Dictionary

Код валюты	Название валюты
BTC	Bitcoin cryptocurrency
LTC	Litecoin cryptocurrency
USD	American dollar
EUR	Euro
RUB	Russian ruble
KZT	Kazakhstan tenge
UAH	Ukrainian hryvnia
*	*

* Any currencies can be added.

4.2 Game Codes

A full list of games can be obtained in json format by executing a GET request by specifying operator id in the operator_id parameter:

```
https://int.apiforb2b.com/frontendsrv/apihandler.api?cmd={"api":"ls-games-by-operator-id-get","operator_id":"0"}
```

Getting game icons from JSON:

```
{
  "ico_baseurl": "/game/icons/",
  "groups": [
    {
      "games": [
        {
          ...
          "icons": [
            {
              "ic_h": 221,
              "ic_w": 300,
              "ic_name": "ReelKing_300_221.jpg"
            },
            {
              "ic_h": 220,
              "ic_w": 220,
              "ic_name": "ReelKing_220_220.jpg"
            },
            ...
          ],
          "gm_title": "Reel King"
        },
        ...
      ],
      ...
    }
  ]
}
```

[https://int.apiforb2b.com\[ico baseurl\]\[ic name\]](https://int.apiforb2b.com[ico baseurl][ic name])

Example:

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

* New games are always in development. This list can be expanded at any time.

4.2 Error Codes

Error code	Error description
0	Completed successfully
1	Common mistake
2	*
3	Insufficient funds
4	Token not found
5	User is not found
6	the user is blocked
7	Transaction not found
8	Transaction Expired
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 source error code. In error_description write transaction has already proceeded.

4.3 List of supported languages

Currency code	Currency Name
ru	Russian language
en	English

If you specify a language that is not in the list, English will be displayed.

5. API 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": "Anonimous",
    "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",
    "credit_type": "regular"
  },
  "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 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": 0,
    "user_id": "1",
    "user_nickname": "Anonimous",
    "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",
      "version": 2
    }
  },
  "success": true,
  "api": "do-get-features-user-ingame"
}
```

5.5 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"
}
```

5.6 Update 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",
      "round_win": "150",
      "count": 10,
      "played": 3
    }
  },
  "api": "do-update-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-update-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": 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-end-features-user-ingame"
}
```