

# Operator Integration API documentation

## v2.49

---

This document and its contents are protected under international copyright law.  
Any unauthorized publication, copying, lending or reproduction is strictly prohibited.

- [Changelog](#)
- [Glossary](#)
- [Important Notes](#)
- [Prerequisites](#)
- [Integration Workflow](#)
- [Game and lobby launch](#)
  - [Game launch URL / Lobby launch URL](#)
  - [Game in REAL mode](#)
  - [Game in FUN/DEMO mode](#)
- [Player game session token](#)
- [Gameplay API](#)
  - [Endpoints](#)
  - [X-Autho rization header](#)
  - [X-Operator-Id header](#)
  - [Request and response signing \(hashing\)](#)
  - [Request](#)
  - [Response](#)
  - [Exception handling](#)
- [API commands](#)
  - [authenticate](#)
  - [balance](#)
  - [changebalance](#)
  - [status](#)
  - [cancel](#)
  - [finishround](#)
  - [ping](#)
- [Games and Lobbies list API](#)
- [Business report API](#)
- [GGR limits API \(DEPRECATED!\)](#)
- [Balance limits API](#)
- [Hand history API](#)



VERSION	CHANGES	DATE
2.00	A major API upgrade.	2020-01-03
2.10	X-AuthORIZATION header. Status request parameter token replaced with user_id. Added user_id parameter to Cancel request. Game list API update.	2020-01-20
2.11	Game launch language parameter format changed to IETF language tag with fallbacks. Change balance request round_finished parameter changed to optional. A new game launch parameter deposit_url.	2020-02-01
2.12	Parameter descriptions updates.	2020-02-04
2.13	Game list API updates.	2020-02-10
2.14	New parameter display_name in authenticate method. API request examples updated.	2020-02-19
2.15	Rebranding. Further updated explanations of API request parameters.	
2.16	<ul style="list-style-type: none"> <li>- Added lobby launch parameters to game launch method</li> <li>- Added X-Operator-Id headers for outgoing requests to operators</li> <li>- Parameter lobby_url renamed to home_url in gamelaunch method (for FUN and REAL mode plays)</li> <li>- Fixed token length in authenticate method from 255 to 50 (vivo's token max length is 50) and minimum length to 20.</li> <li>- Added group parameter in gamelaunch method</li> <li>- Added clientName parameter to gamelaunch method</li> <li>- Removed endsession endpoint</li> <li>- Added operator_id under FUN mode play as required parameter</li> <li>- Added example of request and responses for cancel method</li> <li>- Added user_id in the STATUS request response example</li> <li>- Added description for \$status in response hashing section</li> </ul>	2020-07-21
2.17	<ul style="list-style-type: none"> <li>- Added missing user_id param into status call response example</li> <li>- New parameter added to game launch : device = desktop/mobile</li> <li>- Added new section Important Notes</li> </ul>	2020-09-17
2.18	<ul style="list-style-type: none"> <li>- Added lobby_id parameter for Portomaso Live Casino lobby launch</li> <li>- added important note #3 Session check</li> </ul>	2020-09-24
2.19	<ul style="list-style-type: none"> <li>- Fixed evolution lobby launch parameter</li> <li>- Added note #4 Duplicate transaction example</li> <li>- Fixed portomaso lobby launch description</li> <li>- Added example for lobby launch</li> </ul>	2020-10-07
2.20	Added lobby_id parameter for Live168 lobby launch	2020-10-13
2.21	<ul style="list-style-type: none"> <li>- Added STATUS to list of calls where session check should not be enforced in note #3</li> <li>- Edited token description in game launch section, token should be unique for each game launch</li> <li>- Edited game launch description</li> <li>- Fixed OP_21 error code description</li> </ul>	2020-10-27
2.22	Updated evolution lobby ids and descriptions and added 7.1.2 Lobby ID values section	2021-01-20
2.23	- Added ezugi lobby ids in 7.1.2 Lobby ID values section - Updated vivo lobby ids in 7.1.2 Lobby ID values section	2021-01-27
2.24	<ul style="list-style-type: none"> <li>- Removed lobby id list from api doc (section 7.1.2)</li> <li>- Added lobby list method and examples</li> <li>- Section 7.1.1 changed launch methods examples</li> </ul>	2021-04-01
2.25	- Added businessReport api endpoint and examples	2021-05-07
2.26	- Added handHistory api request and examples	2021-05-26
2.27	Updated handHistory api request a) changed request format b) removed rate limit for requests	2021-07-16
2.28	Updated some text	2021-08-10
2.29	Updated some wrong text	2021-08-12

VERSION	CHANGES	DATE
2.30	<ul style="list-style-type: none"> <li>- Added user_id to <a href="#">balance</a> and <a href="#">changebalance</a> requests</li> <li>- Added important note 5</li> </ul>	2021-09-10
2.31	<ul style="list-style-type: none"> <li>- Renamed header X-TT-Operator-Id to X-Operator-Id (Previous header works for backward compatibility)</li> <li>- Added rake field to changebalance WIN requests</li> </ul>	2021-09-29
2.32	Added X-Vendor-Id header in <a href="#">handHistory</a> request	2021-10-07
2.33	Clarified and re-worded things	2021-10-28
2.34	context.history_id in <a href="#">changebalance</a> , history_id in <a href="#">handHistory</a>	2021-11-26
2.35	Removed <a href="#">cancel</a> from "Applies to" column/list for OP_21 error code	2021-12-23
2.36	Added <a href="#">OP_47 (Timeout)</a> error code	2022-02-16
2.37	Added <a href="#">ggrLimits</a> API endpoint	2022-07-28
2.38	<a href="#">Game list API</a> update: fun_mode: <int> field added for each game	2022-09-12
2.39	<a href="#">authenticate</a> added: group: <string> optional field	2022-10-13
2.40	Clarify <a href="#">OP_49</a> usage	2023-05-09
2.41	<a href="#">status</a> : added transaction_type: <string>, transaction_date: <string>, transaction_ts: <string> to request data parameters (no changes to response data required!)	2023-07-13
2.42	<a href="#">changebalance</a> new transaction context parameter context.game_details	2023-08-07
2.43	New command: <a href="#">finishround</a>	2023-09-15
2.44	Deprecated <a href="#">ggrLimits</a> , added <a href="#">balanceLimits</a>	2023-10-11
2.45	Added optional parameter currencies in <a href="#">balanceLimits</a>	2023-11-16
2.46	<a href="#">Game list API</a> update: campaigns: <int> field added for each game	2024-03-05
2.47	New command/endpoint: <a href="#">ping</a>	2024-04-30
2.48	<a href="#">Game list API</a> update: created_at: <string> field added for each game	2024-05-10
2.49	<ul style="list-style-type: none"> <li>- Updated Session check on WIN notice</li> <li>- Updated <a href="#">changebalance</a> token parameter description</li> </ul>	2024-09-12



**Platform:** The game provider, a remote side from your perspective -> OUR SYSTEM

**Operator:** Casino (or any other system) that integrates this API -> YOUR SYSTEM

**Player:** Casino user that plays a game

---

## Important Notes



### 1. OperatorID Header:

Every request/command sent by PLATFORM has the X-Operator-Id header. It contains the operatorID value assigned to each operator.

### 2. Idempotency:

All methods/commands must work idempotently. Meaning, if an identical request is sent 100 times, it must be processed only once.

### 3. Session Check:

WIN, REFUND calls can come after the player's session is already expired. To avoid queue pileups (of requests that are being retried), it is advised that session checking on WIN, REFUND calls is avoided, but enforced on BET type transactions only. WIN and REFUND can have token = null.

### 4. Duplicate transaction:

The requests should be processed idempotently. This means that they should be processed only once.

For this to be the case, it should also be pointed out that idempotency must work on IDENTICAL transactions being retried (meaning: same game id, same amount, etc...). Only if we send a transaction ID with different data (gameid, amount etc...) then the response should be **OP\_40: Duplicate transaction**.

Duplicate transaction error should be triggered only if we send a transactionID with different data.

### 5. Strings restrictions:

None of the STRING type parameters should contain the symbol |. For example: user\_id, token...

---

## Prerequisites



Operator must:

- Provide API base URL for the staging environment (HTTPS is required)
- Provide IP address(es) of the staging environment
- Provide a list of currencies that need to be configured
- Provide a format of user\_id and user\_name that will be used on game launch

PLATFORM must:

- Provide API\_HOST for the staging environment
- Provide the operator\_id for the staging environment
- Provide the secret key for the staging environment
- Whitelist Operator's IP address(es)
- Provide Backoffice access

---

## Integration Workflow



1. Player launches a game.
2. Operator generates a token that is tied to both the **player and the game**.
3. Token and the rest of the [game launch parameters](#) are passed to PLATFORM in a **game launch URL**.
4. PLATFORM makes the [authenticate](#) call to the operator with the token used in game launch URL.
5. Operator responds to incoming [authenticate](#) call from PLATFORM with player data belonging to the token.
6. If operator's [authenticate](#) call response is successful and all data validation is passed, player is redirected to the game. If not, error message is shown in the frontend.
7. Gameplay API calls ([balance](#), [changebalance](#), [status](#) etc.) are made to the operator.



## Game launch URL / Lobby launch URL

Launching a game means showing the URL that is returned by game launch method on a web page in an iframe, new tab, popup or simply redirecting to it directly. Some of the games have specific opening requirements. **Mobile games should NOT be opened in an <iframe>.** Best way to do it, to avoid resizing issues, is to redirect to the returned URL, or open the returned URL in a new tab.

Games can be played in **real mode** (real money) and, when a game supports it, in **fun/demo mode**.

## Game in REAL mode

**NOTE:** The **operator\_id** parameter and **GAME\_LAUNCH\_HOST** values are provided by PLATFORM.

A real mode **game launch URL** is composed as follows:

```
https://GAME_LAUNCH_HOST/?mode=real&game_id={game_id}&token={payer_game_session_token_generated_by_operator}&currency={currency}&language={language}&operator_id={operator_id}&home_url={home_url}
```

A real mode **lobby launch URL** is composed as follows:

```
https://GAME_LAUNCH_HOST/?mode=real&lobby_id={game_specific_lobby_id}&token={payer_game_session_token_generated_by_operator}&currency={currency}&language={language}&operator_id={operator_id}&home_url={home_url}
```

**Instead of game id simply use correct lobby id**

GET parameters

PARAMETER	OPTIONAL	TYPE	NOTES
game_id	No	INT 6	A unique game id provided by PLATFORM.
token	No	STRING 50	<b>A unique token should be generated for each game launch.</b> Token must match the following regex - <code>[A-Za-z0-9]</code> . Minimum 20 characters long. If game session is ended on the operator side, OP_21 Invalid token response should be used when responding to a request with such token.
currency	No	STRING 4	Player's currency. ISO 4217 format, unless it's some custom non-standard currency.
mode	No	STRING 4	Game mode, real in this case.
operator_id	No	STRING 20	Operator ID provided by PLATFORM. It's used for the configuration on PLATFORM side and to identify operator entities in the Backoffice reports.

PARAMETER	OPTIONAL	TYPE	NOTES
language	No	STRING 5	IETF language tag, for example, en-US. If not provided, English will be set by default. If the specified language is not supported in particular game, language will fall back to English. If language parameter will be in other format, an attempt to map it to IETF language tag will be made, but can lead to unexpected results. More information about IETF: <a href="https://datahub.io/core/language-codes">https://datahub.io/core/language-codes</a>
home_url	Yes	STRING 500	An URL to your lobby. Player is redirected to this URL from within the game when the corresponding button/icon is clicked.
deposit_url	Yes	STRING 500	An URL to your deposit page. Player will be redirected to this URL from the game, when he will click on the corresponding button/icon, if it's present in the game.
lobby_id	/		This parameter is used to launch a lobby instead of a game. Please refer to section <a href="#">Games and Lobbies list</a> for more information on lobby_id values.
device	No	STRING	Two possible values for this parameter: desktop or mobile. Used by some gameproviders in order to open the correct GUI based on operator's request. When the operator wants to show the mobile GUI to the player device=mobile should be sent.
group	Yes	ALPHANUMERIC STRING 14	This creates a separate set of configurations to which you can assign a group of players. To fully configure your groups, please contact PLATFORM. NOTE: This parameter works only for certain vendors!
clientName	Yes	STRING 14	This parameter is used to customize games. For more information please contact PLATFORM. NOTE: This parameter works only for certain vendors!

Real mode **game launch URL** example:

```
https://GAME_LAUNCH_HOST/?operator_id=
examplecasino&mode=real&game_id=37&token=IfpdCsglTNR4xuFH0ijZ&currency=EUR&lan
uage=en&home_url=https%3A%2F%2Fapi-host.com%2Fgeneric%2Fgames
```

Real mode **lobby launch URL** example:

```
https://GAME_LAUNCH_HOST/?operator_id=
examplecasino&mode=real&lobby_id=portomaso&device=mobile&token=IfpdCsglTNR4xuF
0ijZ&currency=EUR&language=en&home_url=https%3A%2F%2Fapi-host.com%2Fgeneric%2F
ames
```

Game in FUN/DEMO mode

**NOTE:** Games launched in fun/demo mode do not communicate with the operator API.

A fun mode game launch URL is composed as follows:

```
https://GAME_LAUNCH_HOST/?mode=fun&game_id={game_id}&currency={currency}&language={language}&lobby_url={lobby_url}
```

PARAMETER	OPTIONAL	TYPE	NOTES
operator_id	No	STRING 20	Operator ID provided by PLATFORM.
mode	No	STRING 4	Game mode, fun in this case (fun/demo mode).
game_id	No	INT 6	A unique game id provided by PLATFORM.
language	No	STRING 5	IETF language tag, for example, en-US. If not provided, English will be set by default. If the specified language is not supported in particular game, language will fall back to English. If language parameter will be in other format, an attempt to map it to IETF language tag will be made, but can lead to unexpected results. More information about IETF: <a href="https://datahub.io/core/language-codes">https://datahub.io/core/language-codes</a>
home_url	Yes	STRING 500	An URL to your lobby. Player is redirected to this URL from within the game when the corresponding button/icon is clicked.
device	No	STRING	Two possible values for this parameter: desktop or mobile. Used by some gameproviders in order to open the correct GUI based on operator's request. When the operator wants to show the mobile GUI to the player device=mobile should be sent.

Fun mode **game launch URL** example:

```
https://GAME_LAUNCH_HOST/?operator_id=examplecasino&mode=fun&game_id=37&language=en
```

## Player game session token



Each game launch must have a **unique** token string passed along. Token should be directly linked to the player and game id, because [authenticate](#) request/command must return **player data** associated with that token.

Token generation implementation is up to you, as long as it is a **STRING** of **20-50 [A-Za-z0-9]** characters. If a game session is ended on the operator side, OP\_21 Invalid token response should be used (when responding to a request containing such a token).





## Endpoints

**Operator MUST create the following API endpoints:**

```
https://OPERATOR_API_HOST/your/chosen/path/to/api/authenticate
https://OPERATOR_API_HOST/your/chosen/path/to/api/balance
https://OPERATOR_API_HOST/your/chosen/path/to/api/changebalance
https://OPERATOR_API_HOST/your/chosen/path/to/api/status
https://OPERATOR_API_HOST/your/chosen/path/to/api/cancel
https://OPERATOR_API_HOST/your/chosen/path/to/api/finishround # optional
https://OPERATOR_API_HOST/your/chosen/path/to/api/ping # optional
```

**STAGING and PRODUCTION endpoints MUST use SSL.**

## X-Authorization header

Every request sent by the PLATFORM contains the X-Authorization header that can be validated by computing the SHA1 hash using the request's command and the secret key.

```
$expectedHeader = sha1($command . $secretKey);
```

If the X-Authorization header is missing or the SHA1 hash does not match, HTTP 403 Access denied response must be returned.

**X-Authorization header validation example**

```
$authorizationHeader = '59e047104b6da4e4f37726af502ff4b1601dc5f8';
$command = 'status';
$secretKey = 'secret_key';

$expectedHeader = sha1('status' . 'secret_key');

if ($authorizationHeader !== $expectedHeader) {
    // TODO: Add HTTP 403 response
    exit();
}
```

## X-Operator-Id header

Each request sent by the PLATFORM contains the X-Operator-Id header, which includes the operatorID value.

This makes it easier for the operator to know whose operatorID each request belongs to (when a casino has more operatorIDs).

## Request and response signing (hashing)

Every request must be validated/checked based on the incoming hash sent with the request body. Every

response must have it's own hash generated.  
SHA1 is used to generate the hashes.

**Request signature/hash** is made using API **command**, **request timestamp** and **secret key** values.

```
$hash = sha1($command . $requestTimestamp . $secretKey);
```

**Request signature/hash example**

```
$secretKey = 'secret_key';  
$hash = sha1('authenticate' . '2020-12-01 13:44:54' . 'secret_key');
```

Response signature/hash is made using **response status**, **response timestamp** and **secret key** values.

```
$hash = sha1($status . $responseTimestamp . $secretKey);
```

**Response signature/hash example**

```
$secretKey = 'secret_key';  
$hash = sha1('OK' . '2020-12-01 15:33:11' . 'secret_key');
```

**NOTE:** \$status can be OK or ERROR

---

## Request

Content-Type: application/json; charset=utf-8.

All requests are HTTP POST requests performed over HTTPS (except [/ping](#), which is a GET).  
request\_timestamp is in UTC time zone.

**Request structure**

```
{  
  "command": "<command>",  
  "request_timestamp": "<Y-m-d H:i:s>",  
  "hash": "<hash>",  
  "data": {  
    "<parameter>": "<value>",  
    "<parameter>": <value>,  
  }  
}
```

---

## Response

Content-Type: application/json; charset=utf-8.

Every response (except [/ping](#)) must pass back the incoming request, set status to OK or ERROR, set response\_timestamp to the moment when sending the response (in UTC time zone) and generate a hash (response signature).

**Successful response structure**

```

{
  "request": {
    "command": "<command>",
    "data": {
      "<request parameter>": "<value>",
      "<request parameter>": <value>
    },
    "request_timestamp": "<Y-m-d H:i:s>",
    "hash": "<request hash>"
  },
  "response": {
    "status": "OK",
    "response_timestamp": "<Y-m-d H:i:s>",
    "hash": "<response hash>",
    "data": {
      "<response parameter>": "<value>",
      "<response parameter>": <value>
    }
  }
}

```

**NOTE:** Response HTTP code MUST be 200. Any other HTTP response code will be considered equal to an error response with error code OP\_50 – Internal error.

## Exception handling

In case of an error, or if there are other obstacles to successfully process a request, a properly formatted error response must be returned.

### Unsuccessful response structure

```

{
  "request": {
    "command": "<command>",
    "request_timestamp": "<Y-m-d H:i:s>",
    "hash": "<request hash>",
    "data": {
      "<request parameter>": "<value>",
      "<request parameter>": <value>
    }
  },
  "response": {
    "status": "ERROR",
    "response_timestamp": "<Y-m-d H:i:s>",
    "hash": "<response hash>",
    "data": {
      "error_code": "<error code>",
      "error_message": "<error message>",
      "<optional response parameter>": "<value>"
    }
  }
}

```

### **NOTES:**

- Response HTTP code MUST be 200. Any other HTTP response code will be considered as equal to

error response with error code OP\_50 – Internal error.

- Error code must be in exact case and format as listed.
- Any unlisted or malformed error code will be treated as OP\_50 – Internal error.
- Error message can be in free form.
- Additional parameters/data can be added in error response field if/when necessary for easier debugging.
- API error log is accessible by operator in the PLATFORM Backoffice for debugging purposes and it's highly recommended to check it during the development phase to make sure that API responses are processed correctly. Log is archived after one week.

ERROR CODE	REASON	APPLIES TO	EXPLANATION AND USE CASES
OP_20	Invalid hash	authenticate balance changebalance status cancel finishround	If you validate a hash of the request and it does not match.
OP_21	Invalid token	authenticate balance changebalance	You should respond with this error code in cases when requests are done with token that has been expired or is unknown to you. This is an absolutely normal response code on player game session expiration and it will be handled accordingly on Platform side.
OP_22	Authorization failed	authenticate	If for any reason game launch cannot be authorized.
OP_30	Invalid currency	balance changebalance	The transaction currency does not match the player currency (or any other currency-related error).
OP_31	Insufficient funds	changebalance	Player does not have enough funds to cover the bet.
OP_32	Limits reached	changebalance	Player has reached any responsible gambling limits.
OP_33	Player is blocked	balance changebalance	Player is blocked and is not allowed to play.
OP_34	Player is not found	balance changebalance status cancel finishround	No player found by user id.
OP_35	Game is disabled	changebalance	This error code should be used in case if for some reason Operator has disabled a particular game.
OP_40	Duplicated transaction ID	changebalance	This error code should be used in case if change balance request comes with transaction id that already is used in another transaction (for different game id, amount etc.). Repeated changebalance requests with identical data should not be considered as an error, since that can happen when failed requests are retried.
OP_41	Transaction not found	status cancel finishround	Requested transaction (or round) id is not found in the Operator's database.

ERROR CODE	REASON	APPLIES TO	EXPLANATION AND USE CASES
OP_47	Timeout	authenticate balance changebalance status cancel finishround	Indicate that the result of the command is unknown because a timeout occurred on your side. When such a response is sent for a BET changebalance call it means that the BET will be canceled.
OP_49	Operation failed	changebalance cancel finishround	This error code should be used when for some reason it is impossible to execute the command (and it would never become possible in the future either!). Something like "casino went bankrupt". Requests with this response error code will not be retried. On cancel, respond OP_41 when you do not have (and never will) the transaction being canceled.
OP_50	Internal error	authenticate balance changebalance status cancel finishround	Default error code that should be used if any other error code does not match the situation. This includes any internal error(s) that may occur (database errors etc.). Any request whose response uses this error code will be retried. Retry logic/behavior varies per vendor: some use cancel, others retry the changebalance call with the same transaction ID.



## authenticate

This is the first call from our server to your server when the player is launching the game. Based on the data in the response, a player account will be created on Platform side.

### Request data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
token	STRING 50	No	Token provided in game launch URL.

### Successful response data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
user_id	STRING 70	No	Player's unique id.
user_name	STRING 255	No	Player's username.
user_country	STRING 100	No	Player's country.
display_name	STRING 100	Yes	Player's name to display in games, for example, Live roulette. If not provided, user_name will be used instead. Note that this parameter should not contain any sensitive information, because it will be publicly visible in the game/table.
group	STRING 100	Yes	Player group
currency_code	STRING 4	No	Player's currency. ISO 4217 format.
balance	DECIMAL 18,2	No	Player's balance.

### An example of an authenticate request from PLATFORM to operator:

```
{
  "command": "authenticate",
  "request_timestamp": "2020-12-01 13:44:54",
  "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
  "data": {
    "token": "eqkB1WoKQCvjrvV6taJRN"
  }
}
```

### An example of an OK authenticate response from operator to PLATFORM:

```
{
  "request": {
    "command": "authenticate",
    "request_timestamp": "2020-12-01 13:44:54",
    "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
    "data": {
      "token": "eqkB1WoKQCvjrV6taJRN"
    }
  },
  "response": {
    "status": "OK",
    "response_timestamp": "2020-12-01 13:44:55",
    "hash": "b456f090dffed71866c0ef18071c40058a173098",
    "data": {
      "user_id": "25343",
      "user_name": "john_doe",
      "display_name": "John Doe",
      "user_country": "Germany",
      "balance": 420.76,
      "currency_code": "EUR"
    }
  }
}
```

An example of an **ERROR** authenticate response from operator to PLATFORM

```
{
  "request": {
    "command": "authenticate",
    "request_timestamp": "2020-12-01 13:44:54",
    "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
    "data": {
      "token": "eqkB1WoKQCvjrV6taJRN"
    }
  },
  "response": {
    "status": "ERROR",
    "response_timestamp": "2020-12-01 13:44:55",
    "hash": "b456f090dffed71866c0ef18071c40058a173098",
    "data": {
      "error_code": "OP_21",
      "error_message": "Invalid token"
    }
  }
}
```

# balance

Returns balance of the player.

## Request data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
token	STRING 50	No	Token provided on game launch.
currency_code	STRING 4	No	The currency code for the expected balance.
user_id	STRING 70	No	User id provided in authenticate response. Must be unique.

## Successful response data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
balance	DECIMAL 18,2	No	Player's balance.
currency_code	STRING 4	No	Player's currency. <b>ISO 4217</b> format.

## An example of a balance request from PLATFORM to operator

```
{
  "command": "balance",
  "request_timestamp": "2020-12-01 13:44:54",
  "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
  "data": {
    "token": "eqkB1WoKQCvj rV6taJRN",
    "user_id": "611",
    "currency_code": "EUR"
  }
}
```

## An example of an OK balance response from operator to PLATFORM



```

{
  "request": {
    "command": "balance",
    "request_timestamp": "2020-12-01 13:44:54",
    "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
    "data": {
      "token": "eqkB1WoKQCvjrV6taJRN",
      "user_id": "611",
      "currency_code": "EUR"
    }
  },
  "response": {
    "status": "OK",
    "response_timestamp": "2020-12-01 13:44:55",
    "hash": "b456f090dffed71866c0ef18071c40058a173098",
    "data": {
      "balance": 327.19,
      "currency_code": "EUR"
    }
  }
}

```

An example of an ERROR balance response from operator to PLATFORM

```

{
  "request": {
    "command": "balance",
    "request_timestamp": "2020-12-01 13:44:54",
    "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
    "data": {
      "token": "eqkB1WoKQCvjrV6taJRN",
      "user_id": "611",
      "currency_code": "EUR"
    }
  },
  "response": {
    "status": "ERROR",
    "response_timestamp": "2020-12-01 13:44:55",
    "hash": "b456f090dffed71866c0ef18071c40058a173098",
    "data": {
      "error_code": "OP_30",
      "error_message": "Invalid currency"
    }
  }
}

```

# changebalance

Change balance request is used to process BET, WIN and REFUND transactions.

## Request data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
token	STRING 50   NULL	No	Token. Can be null.
user_id	STRING 70	No	User id provided in authenticate response. Must be unique.
transaction_type	STRING 20	No	Transaction type. Possible values: BET, WIN, REFUND.
transaction_id	INT 20	No	PLATFORM unique transaction id.
round_id	INT 20	No	PLATFORM unique round id.
round_finished	BOOL   NULL	No	Is PLATFORM round finished. For some game vendors this parameter can be null.
game_id	INT 6	No	PLATFORM game id.
currency_code	STRING 4	No	Player's currency. ISO 4217 format.
amount	DECIMAL 18,2	No	Transaction amount.
transaction_timestamp	DATETIME	No	Timestamp of the transaction creation in UTC time zone in Y-m-d H:i:s format.
context	OBJECT	No	This object can have multiple members. Members vary for different games/game-types and/or vendors.
context.reason	STRING 32	No	Transaction reason. For example, COLLECT, BONUS-GAME, SPIN - can vary per game. If vendor does not provide this information, this parameter will be set to "not_provided". Promotional freespins have the value PROMO-FREESPIN.
context.rake	DECIMAL 18,4	Yes	Poker rake amount sent with a WIN type transaction. It has 4 decimal places. Rake is used for Poker accounting.
context.history_id	STRING	No	History id, required for <a href="#">Hand history API</a> calls.
context.game_details	OBJECT	Yes	Game details field is optional and provides additional information about the game being played.

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
context.game_details.ticket_id	INT 20	<b>Yes</b>	Unique identifier associated with a user's participation in lottery games.
context.campaign_code	STRING	<b>Yes</b>	Campaign code, uniquely representing the created/granted campaign.
context.campaign_details	OBJECT   ARRAY   STRING	<b>Yes</b>	Campaign details (total/remaining spins etc.). Varies in structure/naming currently based on what different vendors support/send.

#### Successful response data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
currency_code	STRING 3	<b>No</b>	Player's currency. <b>ISO 4217</b> format.
balance	DECIMAL 18,2	<b>No</b>	Player's balance after request is processed.

#### An example of a changebalance request from PLATFORM to operator

```
{
  "command": "changebalance",
  "request_timestamp": "2020-12-01 13:44:54",
  "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
  "data": {
    "transaction_type": "WIN",
    "reason": "not_provided",
    "amount": 17.25,
    "currency_code": "EUR",
    "transaction_id": 3568,
    "transaction_timestamp": "2020-12-01 13:44:54",
    "round_id": 1727,
    "round_finished": false,
    "game_id": 200,
    "user_id": "611",
    "token": "eqkB1WoKQCvjrvV6taJRN",
    "context": {
      "reason": "not_provided",
      "rake": 0.0328
    }
  }
}
```

#### An example of an OK change balance response from operator to PLATFORM

```

{
  "request": {
    "command": "changebalance",
    "request_timestamp": "2020-12-01 13:44:54",
    "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
    "data": {
      "transaction_type": "WIN",
      "amount": 17.25,
      "currency_code": "EUR",
      "transaction_id": 3568,
      "transaction_timestamp": "2020-12-01 13:44:54",
      "round_id": 1727,
      "round_finished": false,
      "game_id": 200,
      "user_id": "611",
      "token": "eqkB1WoKQCvjrv6taJRN",
      "context": {
        "reason": "not_provided",
        "rake": 0.0328
      }
    }
  },
  "response": {
    "status": "OK",
    "response_timestamp": "2020-12-01 13:44:55",
    "hash": "b456f090dffed71866c0ef18071c40058a173098",
    "data": {
      "balance": 327.19,
      "currency_code": "EUR"
    }
  }
}

```

■ An example of an ERROR change balance response from operator to PLATFORM

```
{
  "request": {
    "command": "changebalance",
    "request_timestamp": "2020-12-01 13:44:54",
    "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
    "data": {
      "transaction_type": "BET",
      "amount": 17.25,
      "currency_code": "EUR",
      "transaction_id": 3568,
      "transaction_timestamp": "2020-12-01 13:44:54",
      "round_id": 1727,
      "round_finished": false,
      "game_id": 200,
      "user_id": "611",
      "token": "eqkB1WoKQCvjrv6taJRN",
      "context": {
        "reason": "not_provided",
        "rake": 0.0328
      }
    }
  },
  "response": {
    "status": "ERROR",
    "response_timestamp": "2020-12-01 13:44:55",
    "hash": "b456f090dffed71866c0ef18071c40058a173098",
    "data": {
      "error_code": "OP_31",
      "error_message": "Insufficient funds"
    }
  }
}
```

## status

This command is used to get the status of a transaction in the Operator's database. This call can happen at any time and may be used to get transaction status before retrying failed calls.

Please note that `response.status` and `response.data.transaction_status` are two entirely different (unrelated) fields. `response.status` must be either OK or ERROR (as described [here](#)). When set as ERROR, it means you are returning one of available [error-codes](#) as response.

### Request data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
user_id	STRING 70	No	User id provided in authenticate response. Must be unique.
transaction_id	INT 20	No	PLATFORM transaction ID.
transaction_type	STRING 20	No	PLATFORM transaction type. Possible values: BET, WIN, REFUND.
transaction_date	STRING 10	No	PLATFORM transaction creation date (UTC, in YYYY-MM-DD format).
transaction_ts	STRING 13	No	PLATFORM transaction creation date (unix timestamp).
round_id	INT 20	No	PLATFORM round ID.

### Successful response data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
user_id	STRING 70	No	User id provided in authenticate response. Must be unique.
transaction_id	INT 20	No	PLATFORM transaction ID.
transaction_status	STRING 10	No	Transaction status in the Operator database. Possible values: OK, ERROR, CANCELED.

### An example of a status request from PLATFORM to operator

```
{
  "command": "status",
  "request_timestamp": "2020-12-01 13:44:54",
  "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
  "data": {
    "user_id": "17",
    "transaction_id": 3568,
    "transaction_type": "BET",
    "transaction_date": "2020-11-31",
    "transaction_ts": "1606822496000",
    "round_id": 1225
  }
}
```

### An example of an OK status response from operator to PLATFORM

```

{
  "request": {
    "command": "status",
    "request_timestamp": "2020-12-01 13:44:54",
    "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
    "data": {
      "user_id": "17",
      "transaction_id": 3568,
      "transaction_type": "BET",
      "transaction_date": "2020-11-31",
      "transaction_ts": "1606822496000",
      "round_id": 1225
    }
  },
  "response": {
    "status": "OK",
    "response_timestamp": "2020-12-01 13:44:55",
    "hash": "b456f090dfed71866c0ef18071c40058a173098",
    "data": {
      "user_id": "17",
      "transaction_id": 3568,
      "transaction_status": "OK"
    }
  }
}

```

An example of an **ERROR** status response from operator to PLATFORM

```

{
  "request": {
    "command": "status",
    "request_timestamp": "2020-12-01 13:44:54",
    "hash": "43afab643ee5b94e3b4edbe438f30fa3e97e10c4",
    "data": {
      "user_id": "17",
      "transaction_id": 3568,
      "transaction_type": "BET",
      "transaction_date": "2020-11-31",
      "transaction_ts": "1606822496000",
      "round_id": 1225
    }
  },
  "response": {
    "status": "ERROR",
    "response_timestamp": "2020-12-01 13:44:55",
    "hash": "b456f090dfed71866c0ef18071c40058a173098",
    "data": {
      "error_code": "OP_41",
      "error_message": "Transaction not found"
    }
  }
}

```

## cancel

This command is used to cancel transactions. When a transaction is canceled, it must be excluded from the reports and the amount returned to the player (or taken from him, depending on `transaction_type`).

Based on `round_finished` flag the round should be updated.

### Request data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
<code>user_id</code>	STRING 70	No	User id provided in authenticate response. Must be unique.
<code>transaction_id</code>	INT 20	No	PLATFORM transaction ID.
<code>round_id</code>	INT 20	No	PLATFORM round ID.
<code>round_finished</code>	BOOL   NULL	No	Is PLATFORM round finished. For some game vendors this parameter can be null.
<code>game_id</code>	INT 6	No	PLATFORM game ID.

### Successful response data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
<code>user_id</code>	STRING 70	No	User id provided in authenticate response. Must be unique.
<code>transaction_id</code>	INT 20	No	PLATFORM transaction ID.
<code>transaction_status</code>	STRING 10	No	Transaction status in the Operator database. Possible value: CANCELED.

### An example of a `cancel` request from PLATFORM to operator

```
{
  "command": "cancel",
  "data": {
    "user_id": "113",
    "transaction_id": 327128,
    "round_id": 145658,
    "round_finished": null,
    "game_id": 5523
  },
  "request_timestamp": "2020-07-29 07:21:16",
  "hash": "f9267174537b8f279561e0b04cc36120b160ff88",
  "debug": {
    "code": 200,
    "time": "0.04"
  }
}
```

### An example of an OK cancel response from operator to PLATFORM



```

{
  "request": {
    "command": "cancel",
    "data": {
      "user_id": 113,
      "transaction_id": 327128,
      "round_id": 145658,
      "round_finished": null,
      "game_id": 5523
    },
    "request_timestamp": "2020-07-29 07:21:16",
    "hash": "f9267174537b8f279561e0b04cc36120b160ff88"
  },
  "response": {
    "status": "OK",
    "response_timestamp": "2020-07-29 07:21:16",
    "hash": "e0027d6d34e457236e56d608f4b1aaa225f5cde1",
    "data": {
      "user_id": 113,
      "transaction_id": 327128,
      "transaction_status": "CANCELED"
    }
  }
}

```

An example of an ERROR cancel response from operator to PLATFORM

```

{
  "request": {
    "command": "cancel",
    "data": {
      "user_id": 113,
      "transaction_id": 327146,
      "round_id": 145666,
      "round_finished": null,
      "game_id": 5523
    },
    "request_timestamp": "2020-07-29 07:28:26",
    "hash": "97c77dcd649afa2e1841166142495f8c119e4aa7"
  },
  "response": {
    "status": "ERROR",
    "response_timestamp": "2020-07-29 07:28:26",
    "hash": "ce6bbead40e8139a5bd3359c7f6f7dac0df65186",
    "data": {
      "error_code": "OP_41",
      "error_message": "Transaction not found"
    }
  }
}

```

## finishround

This command notifies operators that the specified round is considered finished by the PLATFORM (no new transactions).

Failed calls are retried (depends on vendor). Retries can be stopped by responding OP\_49 Operation failed.

NOTE: Currently supported only for certain vendors!

### Request data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
round_id	INT 20	No	PLATFORM round ID.
user_id	STRING 70	No	User id provided in authenticate response. Must be unique.
game_id	INT 6	No	PLATFORM game ID.
round_date	STRING 10	No	PLATFORM round creation date (UTC, in YYYY-MM-DD format).
round_ts	STRING 13	No	PLATFORM round creation date (unix timestamp).

### Successful response data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
round_id	INT 20	No	PLATFORM round ID.
user_id	STRING 70	No	User id provided in authenticate response. Must be unique.
game_id	INT 6	No	PLATFORM game ID.

### An example of a finishround request from PLATFORM to operator

```
{
  "command": "finishround",
  "data": {
    "round_id": 533281,
    "user_id": "398",
    "game_id": 46,
    "round_date": "2023-09-14",
    "round_ts": "1694704880325"
  },
  "request_timestamp": "2023-09-15 22:22:38",
  "hash": "f3dde71cb8673ce5c894dd50309859192a68e5c8"
}
```

### An example of an OK finishround response from operator to PLATFORM

```

{
  "request": {
    "command": "finishround",
    "data": {
      "round_id": 533281,
      "user_id": "398",
      "game_id": 46,
      "round_date": "2023-09-14",
      "round_ts": "1694704880325"
    },
    "request_timestamp": "2023-09-15 22:22:38",
    "hash": "f3dde71cb8673ce5c894dd50309859192a68e5c8"
  },
  "response": {
    "status": "OK",
    "response_timestamp": "2023-09-15 22:22:38",
    "hash": "db9529772807b8d86bff357b0be895af59dbeeac",
    "data": {
      "round_id": 533281,
      "user_id": "398",
      "game_id": 46
    }
  }
}

```

An example of an **ERROR** finishround response from operator to PLATFORM

```

{
  "request": {
    "command": "finishround",
    "data": {
      "round_id": 533281,
      "user_id": "398",
      "game_id": 46,
      "round_date": "2023-09-14",
      "round_ts": "1694704880325"
    },
    "request_timestamp": "2023-09-15 22:48:01",
    "hash": "f13fb82dfacad45722d83275b31bf057efc885fb"
  },
  "response": {
    "status": "ERROR",
    "response_timestamp": "2023-09-15 22:48:01",
    "hash": "fe86e9280eee4040c7283986c2b6148684398e5f",
    "data": {
      "error_code": "OP_50",
      "error_message": "Internal error"
    }
  }
}

```

## ping

This (optional) request is used for monitoring and debugging basic connectivity/performance issues.

**I** GET request.

Response status must be 200. Response body can be empty (or "PONG").

---

# Games and Lobbies list API



This API provides the ability to automatically fetch games list. Details can vary per game and can be frequently updated. By using this API it's also possible to check for disabled games.

## Authorization

GET request needs to provide custom headers.

HEADER	TYPE	OPTIONAL	DESCRIPTION
X-Authorization	STRING	No	SHA1 hash sha1('games' . \$operatorId . \$secretKey)
X-Operator-Id	STRING	No	Operator ID provided by PLATFORM.

## API endpoint: Games List

```
PLATFORM_API_HOST/api/generic/games/list/all
```

Game list API request example from operator to PLATFORM

```
GET https://api-host.com/api/generic/games/list/all
```

API response structure

```
{
  "games": [
    {
      "id": "<game id>", # integer
      "title": "<game title>", # string
      "platform": "<platform>", # string
      "type": "<game type>", # string
      "enabled": "<1|0>", # integer
      "fun_mode": "<1|0>", # integer, indicates fun/demo mode support per game
      "campaigns": "<1|0>", # integer, indicates campaigns support per game
      "vendor": "<vendor>", # string
      "created_at": "<Date Y-m-d format>", # string
      "details": {
        "description": "...", # string
        "some_other_key": "..." # mixed
      }
    }
  ]
}
```

An example of a successful response from PLATFORM to operator

```
{
  "games": [
    {
      "id": 1,
      "title": "Apocalypse Slot",
      "platform": "desktop",
      "type": "slots",
      "enabled": 1,
      "fun_mode": 1,
      "vendor": "test1",
      "created_at": "2024-05-10",
      "details": {
        "description": {
          "en": "A dark knight rides a horse.",
          "ru": "Темный рыцарь скачет на лошади."
        }
      }
    }
  ]
}
```

#### API endpoint: Lobbies List

HEADER	TYPE	OPTIONAL	DESCRIPTION
X-Authorization	STRING	No	SHA1 hash sha1('lobbies' . \$operatorId . \$secretKey)
X-Operator-Id	STRING	No	Operator ID provided by PLATFORM.

PLATFORM\_API\_HOST/api/generic/lobbies/list/all

#### Lobbies list API request example from operator to PLATFORM

GET https://api-host.com/api/generic/lobbies/list/all

#### NOTES:

- When lobby\_id is provided, game\_id is ignored.
- **To launch Portomaso lobby:** use ?lobby\_id=portomaso with desired device=mobile/desktop parameter
- **To launch Live168 lobby :** use ?lobby\_id = live168 but be aware that this **product has to be opened in a new tab in order to work.**

#### API response structure

```
{
  "lobbies": [
    {
      "lobby_id": "xxxx_game_shows",
      "vendor": "xxxx",
      "details": {
        "thumbnails": {
          "300x300": "https://mediaapihost.com/359/xxxx/300x300/lobb
pic.png"
        }
      }
    }
  ]
}
```

**An example of a successful response from PLATFORM to operator**

```
{
  "lobbies": [
    {
      "lobby_id": "evolution_game_shows",
      "vendor": "evolution",
      "details": {
        "thumbnails": {
          "300x300": "https://mediaapihost.com /359/evolution/300x300
/evolution_game_shows.png"
        }
      }
    },
    {
      "lobby_id": "evolution_baccarat_sicbo",
      "vendor": "evolution",
      "details": {
        "thumbnails": {
          "300x300": "https://mediaapihost.com/359/evolution/300x300
evolution_baccarat_sicbo.png"
        }
      }
    }
  ]
}
```



This API endpoint provides operators the ability to download aggregated operator data.

## Authorization

GET request needs to provide custom headers.

HEADER	TYPE	OPTIONAL	DESCRIPTION
X-Authorization	STRING	No	SHA1 hash sha1('businessReport' . \$operatorId . \$secretKey)
X-Operator-Id	STRING	No	Operator ID provided by PLATFORM.

## Request data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
date	STRING	No	Format: yyyy-mm-dd

## API endpoint: Business report

```
PLATFORM_API_HOST/api/generic/businessReport
```

## Business report API request example from operator to PLATFORM

```
GET https://api-host.com/api/generic/businessReport?date=2021-05-03
```

## An example of a successful response from PLATFORM to operator



```
[
  {
    "entity_name": "operatorName",
    "vendor": "slotsXX",
    "currency_code": "EUR",
    "bets_sum": 6.1,
    "bets_count": 1,
    "wins_count": 0,
    "wins_sum": 3.3,
    "freespins_sum": 0,
    "freespins_count": 0,
    "jackpots_count": 0,
    "jackpots_sum": 0,
    "gross_gaming_revenue": "2.80",
    "date": "2021-05-03"
  },
  {
    "entity_name": " operatorName",
    "vendor": "netent",
    "currency_code": "EUR",
    "bets_sum": 1.2,
    "bets_count": 7,
    "wins_count": 0,
    "wins_sum": 0.31,
    "freespins_sum": 0,
    "freespins_count": 0,
    "jackpots_count": 0,
    "jackpots_sum": 0,
    "gross_gaming_revenue": "0.89",
    "date": "2021-05-03"
  }
]
```

---

# GGR limits API (DEPRECATED!)



Superseded by </api/generic/balanceLimits>. This one is going away!

## Authorization

GET request needs to provide custom headers.

HEADER	TYPE	OPTIONAL	DESCRIPTION
X-Authorization	STRING	No	SHA1 hash sha1('ggrLimits' . \$operatorId . \$secretKey)
X-Operator-Id	STRING	No	Operator ID provided by PLATFORM.

## API endpoint: GGR Limits

```
PLATFORM_API_HOST/api/generic/ggrLimits
```

## GGR Limits API request example from operator to PLATFORM

```
GET https://api-host.com/api/generic/ggrLimits
```

## An example of a successful response from PLATFORM to operator without limits

```
{
  "enabled": false // Not limited
}
```

## An example of a successful response from PLATFORM to operator with limits

```
{
  "enabled": true,
  "currency": "EUR",
  "limit": 1111111,
  "current_usage": 0,
  "percentage_usage": 0
}
```

# Balance limits API



This API endpoint provides operators the ability to retrieve detailed information about their current spending/limits.

## Authorization

GET request needs to provide custom headers.

HEADER	TYPE	OPTIONAL	DESCRIPTION
X-Authorization	STRING	No	SHA1 hash sha1('balanceLimits' . \$operatorId . \$secretKey)
X-Operator-Id	STRING	No	Operator ID provided by PLATFORM.

Query parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
currencies	STRING	Yes	One or more currency_codes (comma-separated string without space between)

API endpoint: Balance Limits

PLATFORM\_API\_HOST/api/generic/balanceLimits

Balance Limits API request example from operator to PLATFORM

GET https://api-host.com/api/generic/balanceLimits?currencies=GBP,EUR

An example of a successful response from PLATFORM to operator without limits

```
{
  "enabled": false // Not limited
}
```

An example of a successful response from PLATFORM to operator with limits

```
{
  "enabled": true,
  "currency": "USD",
  "available_balance": 107881.02,
  "currencies": {
    "GBP": {
      "available_balance": 86347.97
    },
    "EUR": {
      "available_balance": 99164.23
    }
  },
  "limit_exceeded": false
}
```

# Hand history API



This API endpoint provides operators the ability to request detailed information about a transaction.

## Authorization

GET request needs to provide custom headers.

HEADER	TYPE	OPTIONAL	DESCRIPTION
X-Authorization	STRING	No	SHA1 hash sha1( 'handHistory' . \$operatorId . \$secretKey)
X-Operator-Id	STRING	No	Operator ID provided by PLATFORM.
X-Vendor-Id	STRING	No	Vendor of the game where the transaction was made.

## Request data parameters

PARAMETER	TYPE	OPTIONAL	DESCRIPTION
transaction_id	BIGINT	No	Id of the transaction.
history_id	STRING	No	History id of the transaction. From <a href="#">changebalance</a> call's context.history_id data.

## API endpoint: Hand History

```
PLATFORM_API_HOST /api/generic/handHistory/{transaction_id}/{history_id}
```

## Hand History API request example from operator to PLATFORM for transactionID=111

```
GET https://handhistory.api-host.com/api/generic/handHistory/111/350-a1Basdf
```

NOTE: The response includes a field named `type`. This field can have 2 values, `json` or `url`. Below are two examples, one for each type of response:

## An example of a successful response from PLATFORM to operator for TYPE: `json`

```
{
  "success": true,
  "data": {
    "id": "1682880c76095f979a63d0b4",
    "startedAt": "2021-05-26T05:53:22.908Z",
    "settledAt": "2021-05-26T05:53:44.151Z",
    "status": "Resolved",
    "gameType": "baccarat",
    "table": {
      "id": "oytmvb9m1zysmc44",
      "name": "DNT Baccarat A (851)"
    },
    "dealer": {
      "uid": "tts0r7r_____",
      "name": "ROB_279"
    },
    "result": {
      "banker": {
        "score": 9,
        "hand": [

```

```

        "cards": [
            "4D",
            "5S"
        ]
    },
    "player": {
        "score": 9,
        "cards": [
            "2C",
            "7H"
        ]
    },
    "outcome": "Tie"
},
"participants": [
    {
        "casinoId": "xxx0001",
        "playerId": "1503",
        "screenName": "rtest01",
        "sessionId": "pgttgek4lqqqa5j2168288072f825469ec65e43e",
        "casinoSessionId": "ampleRkj7meIpT0k7MHYc6nTH1622008378",
        "currency": "KRW",
        "bets": [
            {
                "code": "BAC_Player",
                "stake": 3000,
                "payout": 3000,
                "placedOn": "2021-05-26T05:53:41.453Z",
                "description": "Player",
                "transactionId": "621498377245476221"
            }
        ],
        "configOverlays": [],
        "playMode": "RealMoney",
        "channel": "desktop",
        "os": "Windows",
        "device": "Desktop"
    }
],
"outcome": "Tie",
"type": "json"
}

```

An example of a successful response from PLATFORM to operator for TYPE: url

```

{
    "success": true,
    "type": "url",
    "data": "https://backstaging.the-rgs.com/rounds/CG-1133856/eyJ0eXAiOiJKV1Q6LCJhbGciOiJIUzI1NiJ9.eyJyYb3VuZGF9pZCI6IkhNLTExMzY4NTYifQ.8ZNfw5x7-j0YRzxX5h8xRmaiIoq0FY7DmuP9KH2ia4"
}

```

