



T H E B I N G O M U L T I P L A Y E R E X P E R T

Document	E2E GameHub API
Version	1.3.2
Release date	23/08/2022

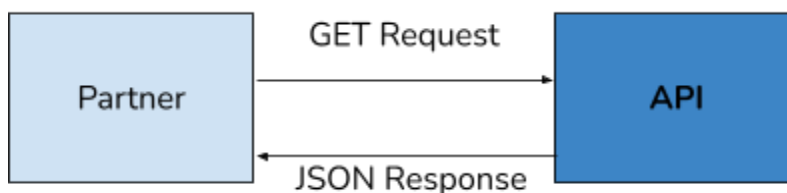
Data Dictionary

GH	GameHub. This is E2E backend utility, which together with the backend functionality provides an API to access a specific set of data and reports.
Partner	This is you, the one that will be using E2E's software.
API	The piece of software this document talks about. GH API provides access to a specific set of data and reports for the partner to use.
API URL	The URL you will be making your requests to consume the API.
Token	A unique identifier for the partner to access the API. It will be provided by E2E, and it can be updated at any moment for security reasons.
Action	The action that will be requested to perform in the API
Action parameters	The parameters that the action expects to receive. In case of the <u>reports</u> action, they will be the report type and the filters for the report.
Report	The report (set of information) that will be consumed by the partner in each request to the API
Filters	The filtering criteria that will be used to fetch the information for the report. Each report will have its own filtering criteria.
Filter type	Expected data type for every filter in the report. The types for the filter will be described further in this document.
Response	The JSON answer for every API call according to the result

Recommendations

- You should store and use the token and other access information in a way that it can easily be changed. We might update the token or the access gateways for security reasons.
- The authentication mechanism might change in the future, so you should have your code prepared to adapt to it.
- The API will always give you a JSON response. Nevertheless, make sure to be prepared to receive corrupted/malformed responses and catch the errors so your application does not break. While errors are not likely to happen, we recommend our end users to be ready to adapt to any bug situations beforehand. We highly recommend you to log/store any type of errors you find while using the service and send them right to us. That way we will be able to fix the issue as soon as possible and improve your experience with our products.

Flow diagram



The API flow is as simple as it gets. You make a GET request with the correct parameters to the API URL (token + action + action parameters), and the API gives you a JSON response (which will be detailed further in this document).

Request

The API expects your GET requests to have the following format:

API_URL / **ACTION** / ?api_token=**TOKEN**&...**parameters**

Where:

API URL	The URL we will provide you to use the API
Token	The unique token we will provide you to use the API
Action	The action that you want to perform in the API
Parameters	The parameters for the action you want to perform in the API

Request Basic Example

For this basic API example we will use the following sample data

API URL	<code>https://127.0.0.1:8000/api/v1/</code>
Token	<code>sampletoken123</code>
Action	<code>reports</code>
Parameters	<ul style="list-style-type: none">• report_type = <code>sample</code>• user_type = <code>1</code>• hive_group_id = <code>100</code>

This means we want to perform the action `reports`, and retrieve the `sample` report, with the parameters user_type = `1`, and hive_group_id = `100`.

This is how our request should look:

`https://127.0.0.1:8000/api/v1/reports/?api_token=sampletoken123&report_type=sample&user_type=1&hive_group_id=100`

Response

The API always gives you a JSON response according to the result of the request. This response may vary for every action. However, all of the responses share a common format and status response type.

HTTP response status	What it means
400	Unexpected error
401	Authentication problems
404	Invalid action, invalid request, malformed response
422	Validation failure or empty response data
200	Response is OK

If any error has been found, the status of the response status will be in the range of 400 statuses. If the operation is successful, the response status will be 200.

Failed response

All error responses contain an error code and a message for you to parse and show the way it suits you better. At the end of this document you can check the **ERROR INDEX** table to handle the errors in your side of the implementation.

Here's an example of JSON response with errors (in this case the token was not valid):

```
{"error":true,"code":"00001","message":"These credentials do not match our records."}
```

error	Boolean field, always TRUE if the request has failed. Besides the response's status code, you can use this field to check the operation result
code	Code of the error returned by the API. You can check the ERROR INDEX to see the reference for this error
message	Message of the error. If the user is authenticated (meaning token is correct) and has a default language supported by the API, the message displayed for each error will have the correct translation.
... variables	Any variables can also be passed in the failed response, and they will be specific for every action.

Successful response

When a request is processed with no errors, you will get a JSON response with the “error” boolean flag in false, and a set of variables depending on the action that was requested.

Here’s an example of a JSON response with no errors and a report:

```
{ "error": false, "report": [ { col1:foo, col2:bar }, {col1:foo2, col2:bar2} ] }
```

error	Boolean field, always FALSE when the response is correct and no errors were found.
... variables	Any variables can be passed in the successful response, and they will be specific for every action.

Common Parameters

The parameters on the request depends on the type of report being requested. The specific **key names** will be detailed in the Reports section.

Key Types	Details
Report Type	Report identifier. This key name is shared between all reports. Ex: ...& report_type=api_movements_external_report &...
Dates	<p>In general, only the time frame name is needed unless stated otherwise. The parameter key name varies between reports. Ex: ...&timestamp=last_month&...</p> <p>There are multiple time frames available:</p> <ul style="list-style-type: none">• today• yesterday• this_week• last_week• this_month• last_month• this_year• range : Two more parameters are required when selecting a date range. They’re composed of the key name plus _range_from or _range_to indicating the beginning and end of

	<p>the time frame. The date format is Y-m-d, or the unix timestamp.</p> <p>Ex: ...&timestamp=range&timestamp_range_from=2019-08-23&timestamp_range_to=1574467200&...</p> <ul style="list-style-type: none"> • datetime_range : Same as the range time frame, this option needs the _range_from and _range_to parameters, but, in addition, it needs two more parameters indicating the beginning and ending hour. _range_from_time and _range_to_time. The hour format is H:i:s <p>Ex:</p> <p>...&starts_at=datetime_range&starts_at_range_from=2019-08-23&starts_at_range_to=2019-08-27&starts_at_range_from_time=09:45:00&starts_at_range_to_time=21:15:30&...</p>
--	---

Reports

The action utilized to get these reports is **reports**.

Ex: **.../api/v1/reports/?api_token=...**

Bingo Financial Report Daily Movements

Report of financial movements separated by day. This include total wagers, paid in prizes, refunds, real paid, tickets paid and net.

Parameters	
report_type	users_report
groupid	Required. Integer. Identifier of the group you want to receive data from. Ex: ...& groupid=23&...
timestamp	Required. Date.

Response	
timestamp	Date of the movements grouped
group_id	Group of the movements grouped
external_group_id	External ID of group of the movements grouped

group_name	Description name of group
total_wagers_real	Total of real wagers in that date
total_wins_real	Total of prizes won on that date
total_refunds_real	Total refunds on that date
total_paid	Total of prizes paid on that date
total_paid_tickets	Number of tickets paid on that date
net	Net of Total wagers - Total refunds - Total Pairs

Bingo Financial Report Daily Movements - Example JSON response:

```
{
  "error": false,
  "Report": [
    {
      "timestamp": "2019-09-05",
      "group_name": "Local 1 (0.00%)",
      "group_id": 350,
      "external_group_id": 350,
      "total_wagers_real": "30.00",
      "total_wins_real": "0.00",
      "total_refunds_real": "0.00",
      "total_paid": "-20.00",
      "total_paid_tickets": 5,
      "net": "10.00"
    },
    {
      "timestamp": "2019-09-06",
      "group_name": "Local 1 (0.00%)",
      "group_id": 350,
      "external_group_id": 350,
      "total_wagers_real": "3.00",
      "total_wins_real": "0.00",
      "total_refunds_real": "0.00",
      "total_paid": "0.00",
      "total_paid_tickets": 0,
      "net": "3.00"
    }
  ]
}
```


Bulk Games Report

Bulk report of games from a specific Bingo Room ID, filtered by STATUS and DATE.

Parameters	
report_type	api_bulk_games_report
bs_room_id	Integer. Identifier of the room you want to receive data from. Ex: ...&bs_room_id=2&...
starts_at	Required. Date.
status	<ul style="list-style-type: none">• 1 : Selling• 2 : Playing• 3 : Played• 4 : Not Played• 5 : Cancelled• 6 : Preselling• 7 : Cancelling• 8 : To process cancel
per_page	-1 (unlimited data without pagination)

Response	
id	Game identifier
room_id	Room identifier
game_configuration_id	Configuration identifier
game_configuration_name	Configuration Name
game_configuration_key	Root configuration id. In case it's cloned from a previous configuration. Default 0.
status	Status identifier
status_name	Name of the game status
card_price	Price of the game cards
cards_bought	Numbers of cards bought in the game
wagered	Total money gathered in the game

paid	Amount of money paid to the winners
prizes	Object prizes. Total number of prizes won. If multiple cards won the prize, there will be multiple instances of said prize. <pre>{ "id":3499, "prize_name":"One line", "paid":"11.39", "call_number":37, }, { "id":3500, "prize_name":"Two lines", "paid":"22.65", "call_number":57, }</pre>
winners	Object winners. Winners information. <pre>{ "user_name":"u1", "user_id":60, "external_user_id":60, "card_id":"6010-2", "call_number":73, "card_holder_name":"u1", "ticket_id":0, "prize_name":"One line" }</pre>
starts_at	Date when the game started. Timestamp
finished_at	Date when the game finished. Timestamp

Bulk Games Report - Example JSON response:

```
{
  "error": false,
  "report":
  [{
    "id": 1384860,
    "room_id": 1,
    "room_name": "Example",
    "game_configuration_id": 134,
    "game_configuration_name": "The test",
    "status": 3,
    "status_name": "Played",
    "card_price": "5.00",
    "cards_bought": 24,
    "wagered": "120.00",
    "paid": "81.71",
    "prizes": [
      {
        "id": 3499,
        "prize_name": "One line",
        "paid": "15.10",
```

```

        "call_number": 37
    },
    {
        "id": 3500,
        "prize_name": "Two lines",
        "paid": "22.65",
        "call_number": 57
    },
    {
        "id": 3501,
        "prize_name": "Fullhouse",
        "paid": "37.77",
        "call_number": 68
    },
    {
        "id": 3502,
        "prize_name": "PJP",
        "paid": "6.19",
        "call_number": 68
    }
  ],
  "winners": [
    {
      "user_name": "u1",
      "user_id": "60",
      "card_id": "6010-2",
      "call_number": 73,
      "card_holder_name": "u1",
      "ticket_id": 0,
      "prize_name": "One line"
    },
    {
      "user_name": "u1",
      "user_id": "60",
      "card_id": "6010-1",
      "call_number": 81,
      "card_holder_name": "u1",
      "ticket_id": 0,
      "prize_name": "Two lines"
    },
    {
      "user_name": "u1",
      "user_id": "60",
      "card_id": "6010-1",
      "call_number": 85,
      "card_holder_name": "u1",
      "ticket_id": 0,
      "prize_name": "Fullhouse"
    }
  ],
  "starts_at": 1574351555,
  "finished_at": 1574351633
}]
}

```

Retail Game Details

Does not require a date parameter.

Parameters	
report_type	api_retail_game_details
id	Required. Integer. Identifier of the game you want to receive data from. Ex: ...&id=23&...

Response	
game_id	Game identifier
room_name	Name of the room where the game was played
game_variant	Game variant name
room_id	Room identifier
game_configuration_id	Game configuration identifier
paid	Total money paid from prizes
configuration	Configuration name
configuration_key	Root game configuration identifier. In case it's cloned from a previous configuration. Default is 0.
available_balls	Available balls of game
balls_delay	Seconds between balls draw
jackpot_build_method	Build method. "Fixed": Jackpot does not increase "Wager": Jackpot increases based on wagers
progressive_jackpot	False by default. If progressive jackpot is configured, this returns an object. <pre>{ "name": "PJP Name", "pjp_winnable_until_call": 85, "winners": [{</pre>

	<pre> "ticket_id": "10807-2", "amount_won": "7.29", "user_id": 56, "group_id": 2 }, ...], </pre>
starts_at	Game duration
ends_at	Date when the game finished
players	Number of players who played the game
cards_bought	Numbers of cards bought in the game
wagered	Total money gathered in the game
wagers_to_prizes	Part of total wagered money, gathered in prizes
wagers_to_pjp	Part of total wagered money, gathered in the progressive pool
pjp_percentage	Percentage of (wagers - commissions) rolled to PJP
profit	Wagered - wagers_to_prizes - wagers_to_pjp
card_price	Card price
status_id	<ul style="list-style-type: none"> • 1 : Selling • 2 : Playing • 3 : Played • 4 : Not played • 5 : Cancelled • 6 : Preselling • 7 : Cancelling • 8 : To process cancel
status_name	Game status name
prizes	<p>Object prizes. Total number of prizes won. If multiple cards won the prize, there will be multiple instances of said prize.</p> <pre> { "id": 3676, "prize_name": "One line", "paid": "11.39", "bs_user_bought_strip_id": 10813, "won_at_card": 8, "call_number": 37, "winners": [{ "ticket_id": "10813-4", "amount_won": "11.39", </pre>

	<pre>"user_id":56, "external_user_id":56, "external_group_id":2 }]</pre>
tickets	<p>Tickets bought in the game. Object</p> <pre>[{ "ticket_id":"164522q8bs", "created_at":1574694799, "seller_group_id":2, "external_seller_group_id":2, "win_info":[{ prizes: { "3678": { "id": 3678, "name": "Fullhouse", "amount": "17.84" }, "3679": { "id": 3679, "name": "PJP", "amount": "7.29" } }, "card_id": "10807-2" }], "pay_info":{ "user_id":8241, "group_id":9614 "external_user_id":8241, "external_group_id":9614 }, "sell_info":{ "group_id":2, "user_id":56 "external_user_id":56, "external_group_id":2 }, "cards":[{ "numbers":[4,14,36,58,68,12,21,47,..], "card_id":"10806-1" }, ...] }]</pre>
winners	<p>Object winners. Winners information.</p> <pre>{ "user_name":"u1", "user_id":"60", "external_user_id":"60", "card_id":"6010-2", }</pre>

	<pre> "call_number":73, "card_holder_name":"u1", "ticket_id":0, "prize_name":"One line" } </pre>
balls	<p>Order of balls draw in the game</p> <pre> [84,14,45,43,51,68,17,18,90,7...,] </pre>

Simplified Retail Game Details

Does not require a date parameter.

Parameters	
report_type	api_retail_game_details_simplified
id	Required. Integer. Identifier of the game you want to receive data from. Ex: ...& id=23 &...

Response	
bs_game_id_main	Game identifier
bs_room_id	Name of the room where the game was played
bs_game_variant_description	Game variant name
bs_game_configuration_description	Game configuration name
continuous_game_configuration_description	Continuous Game Configuration name (if is available)
continuous_games	Continuous games count
game_available_balls	Available balls of game

draw_balls_delay	Seconds between balls draw
jackpot_build_method	Build method. "Fixed": Jackpot does not increase "Wager": Jackpot increases based on wagers
jackpot_prize_mode	Jackpot prize mode description
pjp_id	Progressive Jackpot Name
pjp_winnable_until_call_included	Maximum number of ball draws for winning the progressive jackpot
starts_at	Date when the game started
ends_at	Date when the game finished
duration	Duration of game in seconds
total_players	Number of players in the game
total_tickets	Number of cards bought in the game
total_tickets_refunded	Number of tickets refunded
total_wagers_to_prizes_real	Total amount of wagers to real prizes
total_wagers_to_pjp_real	Total amount of wagers to real PJP
net_win_real	Total amount of net win
total_wagers_real	Total amount of wagers
total_refunds_real	Total amount of refunds
card_price	Price of the game cards
status	<ul style="list-style-type: none"> • 1 : Selling • 2 : Playing • 3 : Played • 4 : Not played • 5 : Cancelled • 6 : Preselling • 7 : Cancelling • 8 : To process cancel
bs_game_prizes	Object prizes. Total number of prizes won. If multiple cards

won the prize, there will be multiple instances of said prize.

JSON OUTPUT:

```
[{
  "prize_name": "One line",
  "amount": "0.49",
  "winners": [
    {
      "username": "TEST",
      "user_id": 2090,
      "amount": {
        "real": "0.49",
        "bonus": "0"
      },
      "ticket_id": "7692-6",
      "external_user_id": 4677
    }
  ]
},
{
  "prize_name": "Two lines",
  "amount": "0.66",
  "winners": [
    {
      "username": "TEST",
      "user_id": 2090,
      "amount": {
        "real": "0.66",
        "bonus": "0"
      },
      "ticket_id": "7692-6",
      "external_user_id": 4677
    }
  ]
},
{
  "prize_name": "Fullhouse",
  "amount": "2.17",
  "winners": [
    {
      "username": "TEST",
      "user_id": 2090,
      "amount": {
        "real": "2.17",
        "bonus": "0"
      },
      "ticket_id": "7692-6",
      "external_user_id": 4677
    }
  ]
}]
```

	Order of balls draw in the game JSON OUTPUT: [84,14,45,43,51,68,17,18,90,7...,]
--	--

Retails Movements

Get a report with retail financial movements.

Parameters	
report_type	api_retail_movements
timestamp	Required. Date.
group_id	Required. Integer. Identifier of the group you want to receive data from. Ex: ...& group_id=23 &...

Response	
movement_id	Movement identifier
timestamp	Date when the movement was created
group_id	Identifier of the group (E2E ID)
external_group_id	External ID of the group (your group ID)
group_name	Name of the group
user_id	Identifier of the user who created the movement (E2E ID)
external_user_id	Identifier of the user who created the movement (your group ID)
username	Name of the user
type	Type of the movement.
currency_amount	Game duration
ticket_id	Ticket identifier
game_id	Game identifier

Retail Movements - Example JSON response:

```
{
  "error": false,
  "report": [
    {
      "movement_id": 3500,
      "timestamp": 1660752199,
      "group_id": 390,
      "external_group_id": 390,
      "group_name": "E2E Testing Group",
      "user_id": 2092,
      "external_user_id": 2092,
      "username": "E2E Testing User 2C6F62B",
      "type": "Bingo - Wager",
      "currency_amount": 1,
      "ticket_id": "",
      "game_id": ""
    },
    {
      "movement_id": 3496,
      "timestamp": 1660752018,
      "group_id": 390,
      "external_group_id": 390,
      "group_name": "E2E Testing Group",
      "user_id": 2092,
      "external_user_id": 2092,
      "username": "E2E Testing User 2C6F62B",
      "type": "Bingo - Wager",
      "currency_amount": 3,
      "ticket_id": "",
      "game_id": ""
    }
  ]
}
```

Selling and pre-selling games

Get a report with the games that are currently selling or pre-selling in a specific bingo room (ID).

Parameters	
report_type	api_selling_and_preselling_games_report
starts_at	Required. Date.
bs_room_id	Required. Integer. Identifier of the room you want to receive data from. Ex: ...&bs_room_id=23&...

This response is a collection of games. Each item describes a property of the game.

Response	
Id	Identifier of the game
bs_room_id	Name of the room the game is playing at
starts_at	Date of the start of the game. Timestamp.
cards_bought	Number of cards bought at the game.
canceled_tickets_sum	Total of cancelled tickets in the game.
card_price	Card price
status	Status of the game
prizes	Collection of prizes object. <pre>{ "id":1, "description":"One line", "amount":"11" }</pre>

Selling and pre-selling games - Example JSON response:

```
{
  "error": false,
  "report": [
    {
      "id": 9290776,
      "bs_room_id": "E2E Bingo 90",
      "starts_at": 1660734078,
      "cards_bought": 0,
      "canceled_tickets_sum": 0,
      "card_price": "1.00",
      "status": "Currently Selling",
      "prizes": [
        {
          "id": 1,
          "description": "One line",
          "amount": "0"
        },
        {
          "id": 2,
          "description": "Two lines",
          "amount": "0"
        },
        {
          "id": 3,
          "description": "Fullhouse",
          "amount": "0"
        }
      ]
    },
    {
      "id": 9299270,
      "bs_room_id": "E2E Bingo 90",
      "starts_at": 1660754302,
      "cards_bought": 0,
      "canceled_tickets_sum": 0,
      "card_price": "1.00",
      "status": "Currently Selling",
      "prizes": [
        {
          "id": 1,
          "description": "One line",
          "amount": "0"
        },
        {
          "id": 2,
          "description": "Two lines",
          "amount": "0"
        },
        {
          "id": 3,
          "description": "Fullhouse",
          "amount": "0"
        }
      ]
    }
  ]
}
```

User Transactions Report

Parameters	
report_type	transaction_report
user_id	Required. Integer. Identifier of the user you want to receive data from. Ex: ...& user_id=23 &...
timestamp	Required. Date.
per_page	-1 (unlimited data without pagination)

Response	
movement_id	Identifier of the transaction
group_id	Group of the transaction (E2E group ID)
external_group_id	Group of the transaction (your group ID)
group_name	Description of group
movement_type	Type of the transaction
currency_amount	Total amount of the generated transaction
currency_id	Currency of the transaction

User Transactions Report - Example JSON response:

```
{
  "error": false,
  "report": [
    {
      "movement_id": 3506,
      "timestamp": "2022-08-18 15:28:42",
      "group_name": "E2E Development - Physical",
      "group_id": 382,
      "external_group_id": 382,
      "movement_type": "Bingo - Wager",
      "currency_amount": "-0.10",
      "currency_id": "Real USD",
      "base_currency_amount": "-0.10",
```

```

    "base_currency_id": "Real USD"
  },
  {
    "movement_id": 3505,
    "timestamp": "2022-08-18 15:27:14",
    "group_name": "E2E Development - Physical",
    "group_id": 382,
    "external_group_id": 382,
    "movement_type": "Bingo - Wager",
    "currency_amount": "-0.10",
    "currency_id": "Real USD",
    "base_currency_amount": "-0.10",
    "base_currency_id": "Real USD"
  }
],
}

```

Group List Report

Does not require an id parameter, but if it is provided the report will filter the group with that id.

Parameters	
report_type	api_groups_report
id	Integer. Identifier of the group you want to receive data from. Ex: ...&id=23&...

Response	
id	Identifier of the group
name	Group name
type	Group type

Group List Report - Example JSON response:

```

{
  "error": false,
  "report": [
    {
      "id": 347,
      "external_group_id": 347,
      "name": "Integrations",
      "type": "Manager"
    }
  ]
}

```

User List Report

Does not require any parameter, but if it is provided the report will filter the user with that parameter.

Parameters	
report_type	api_users_report
id	Integer. Identifier of the user you want to receive data from. Ex: ...&id=23&...
username	String, Search in name of the users, may return many results
fkgroup	String, Group id of user, may return many results

Response	
id	Identifier of the user (E2E user ID)
external_user_id	Identifier of the user (your user ID)
username	Name of the user
fkgroup	Group name of user
group_id	Group ID of user (E2E group ID)
external_group_id	Group ID of user (your group ID)

User List Report - Example JSON response:

```
{
  "error": false,
  "report": [
    {
      "id": 23,
      "username": "E2E Testing User",
      "fkgroup": "Group Name"
    }
  ],
  "break": false
}
```


Progressive Jackpots Summary

This report fetches the current state of ALL of the progressive Jackpots being used. Does not require any parameter

Parameters	
report_type	progressive_jackpot_summary

Response [array of]	
description	Name of the progressive jackpot
value	Current VISIBLE value of the PJP
value_max	Max value that can be reached by the Visible PJP
percentage	Percentage of WAGERS to be distributed to the PJP (visible + hidden are included in this)
seed	Current HIDDEN value of the PJP
seed_base	Base value in which the hidden PJP starts from
seed_max	Maximum value that the hidden PJP can reach
seed_percentage	Percentage of “ percentage ” value to be distributed to the hidden PJP (IE 30% of the percentage of wagers to be used to PJP: If the percentage is 5% of \$1000 in wagers, \$50 would be rolled to the PJP in which its 30% -\$15- would go to the hidden and its 70% -\$35- would go to the visible)

Groups Reports

The action utilised to get these reports is **group_reports**.

Ex: **.../api/v1/group_reports/?api_token=...**

Financial Movements Report

Parameters	
report_type	api_movements_external_report
group_id	Integer. Identifier of the group you want to receive data from. Ex: ...&group_id=23&...
timestamp	Required. Date.
per_page	-1 (unlimited data without pagination)

Response	
type	<ul style="list-style-type: none">• 1 : Wager transaction• 3 : Refund transaction• 100 : API transaction
subtype	Type 1, 3: <ul style="list-style-type: none">• 1 : Bingo Transaction Type 3: <ul style="list-style-type: none">• 2: API refund due to cancellation Type 100: <ul style="list-style-type: none">• 2 : Credit Prepaid Bingo Winning• 3 : Credit Ticket Commission• 5 : Credit Ticket Commission Refund
group_id	Group Identifier of the transaction generated. Can be one of their descendant groups.
user_id	User identifier of the generated movement.
game_id	Game identifier where the movement was generated.
currency_amount	Total amount of the generated movement.

Financial Movement Report - Example JSON response:

```
{
  "error":false,
  "report":[
    {
      "timestamp"      : 1567706013,
      "group_id"       : 23,
      "external_group_id" : 23,
      "type"           : 1,
      "subtype"        : 1,
      "currency_amount" : "1.00",
      "user_id"        : 56,
      "external_user_id" : 56,
      "game_id"        : 787048,
      "username"       : "tv1"
    },
    {
      "timestamp"      : 1567709616,
      "group_id"       : 23,
      "external_group_id" : 23,
      "type"           : 1,
      "subtype"        : 1,
      "currency_amount" : "1.00",
      "user_id"        : 56,
      "external_user_id" : 56,
      "game_id"        : 787422,
      "username"       : "tv1"
    }
  ]
}
```

Errors Index

Code	Meaning
00001	User unauthenticated / User unauthorized
10000	Invalid report

10001	Malformed report
10002	Empty data / Failed validation
10003	Jwt error
10011	Invalid group
10012	Invalid date
10013	Invalid user
10014	Invalid room
10015	Invalid status
10016	Invalid game id
10017	Invalid integer
10018	Real time report Break
10019	Invalid chat name