



DRAGONGAMING

CONNECT API DOCUMENTATION

— TRANSFER WALLET —

Contents.

Introduction	5
Glossary	5
Authentication	5
API URLs + API Keys	5
Request and Response	5
Server to Server Communication	6
API Requests from Operator to Provider	6
API Requests Flow	6
• GetGames() Flow	7
• GetGames Request	7
• GetGames Error Response	7
GetGames() API	8
• GetGames() Request	8
• GetGames() Response Example	9
EnterGame() API	10
• EnterGame() Request	10
• EnterGame() Response Example	11
GameExit() API	12
• GameExit() Request	12
• GameExit() Response Example	12

Contents.

CreatePlayer() API	13
• CreatePlayer() Request	13
• CreatePlayer() Response	14
UpdatePassword() API	15
• UpdatePassword() Request	15
• UpdatePassword() Response	15
WalletDeposit() API	16
• WalletDeposit() Request	16
• WalletDeposit() Response	16
WalletWithdraw() API	17
• WalletWithdraw() Request	17
• WalletWithdraw() Response	17
WithdrawStatus() API	18
• WithdrawStatus() Request	18
• WithdrawStatus() Response	18
WalletTransactionStatus() API	19
• WalletTransactionStatus() Request	19
• WalletTransactionStatus() Response	19
WalletGetBalance() API	20
• WalletGetBalance() Request	20
• WalletGetBalance() Response	20

Contents.

PlayerBets() API	21
• PlayerBets() Request	21
• PlayerBets() Response	22
GameHistory() API	23
• GameHistory() Request	23
• GameHistory() Response	24
GameHistoryAllPlayers() API	25
• GameHistoryAllPlayers() Request	25
• GameHistoryAllPlayers() Response	26
GrantBonus() API	27
• GrantBonus() Request	28
• GrantBonus() Response	29



Introduction.

As a game provider, our objective is to deliver exceptional games and experiences to our customers and their players. We would like to see this translate into you integrating our games onto your casino lobby.

This document provides an insight into our API, known as “Chronos”. It addresses the key interactions and requirements from both DragonGaming™ and your perspective. It provides coverage over a wide range of services which can be used by you on your websites, game-servers, and any other API services.

You are required to follow a standard integration process; however, this document will provide a brief overview on several key functions required of our API. This includes integration to Chronos API, API request details, server-to-server integration, game integration and wallet transactions.

We have a separate integration guide that explains the overall integration spec and also includes all the various tools, currencies and languages that we support.

Glossary.

Terms	Description
Operator	Anyone that owns / operates a website / frontend.
RGS	Remote Game Server (by provider)
Provider	DragonGaming™ / Gaming software provider
NG-CONNECT	DragonGaming™ API. RGS-API
Session ID / Token	Session ID of the player
API Key	Unique key provided by DragonGaming to each operator.
Frontend / Site / White-label / Platform	Operator's website in which games are displayed.
Game Lobby	Area where games are shown on operator's website.

The following steps will aid in the integration between DragonGaming and new operators.

Authentication.

- Each API user must obtain a unique API key
- Each request must contain the API key so each API request can be validated.

API URLs + API Keys.

- Any API URL points, and API Keys items will be shared separately with operators.

Request and Response.

- HTTP Header “Content-Type”: “application/json”
- Data sent in POST request should be in JSON format
- All the API responses are in JSON format



Server to Server Communication.

Server-level communications between provider and operator would be conducted in JSON format and is transmitted over HTTP. The current default allows one request at any one time; however, simultaneous requests are possible if required. By default, all currency values are provided in cents with no decimal values. In exceptional cases, these can be provided in a base currency (Pounds/Euros).

API Requests from operator to Provider.

The following examples are some API requests that are currently provided to operators and third parties. This non-exhaustive list is provided as a brief overview. The number and type of requests can increase, as per an operator's requirements.

API Requests Flow .

Following would be the typical flow of few of the API requests.

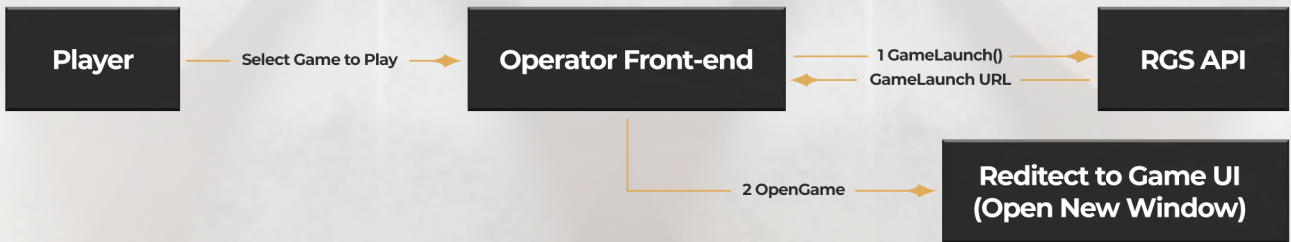
- CasinoSite (front-end of the Operator) is the example website name.
- Player A visits and logs into CasinoSite by providing Username and Password.
- CasinoSite verifies Username and Password match and performs typical checks. If everything is fine, a unique session ID (token) is created for the player in CasinoSite platform.
- Player A visits the game lobby.
- CasinoSite makes the API call GetGames() to Chronos to get the list of games enabled for the CasinoSite. Chronos replies with the list of games enabled.
- List of games are shown to the player in game lobby.
- Player A clicks on any of DragonGaming™ games. CasinoSite needs to make GameLaunch() API call to Chronos API by providing the required details as part of API request.
- Chronos registers the session ID and creates player details if not present and ignores if already available in RGS platform and returns game launch URL to CasinoSite.
- Game is launched (in a new window or redirected to the URL) in CasinoSite with the given GameLaunch URL sent by Chronos.
- Player A plays the game which results in debits and credits. RGS game engine makes debit and credit calls to Chronos accordingly.
- The API Chronos, realizing that the player's wallet is hosted in CasinoSite platform, redirects the calls to the CasinoSite API platform. CasinoSite designs and implements this API and performs the operations on their account wallets. In return, CasinoSite sends player's balance details post debit/credit.
- During every call, API_KEY is passed to Chronos by CasinoSite API.
- During all these calls, the session ID is passed to Chronos by CasinoSite and same session ID will be sent back to CasinoSite during player transactions API calls. CasinoSite recognizes the provided session ID and performs the operations on their wallet.



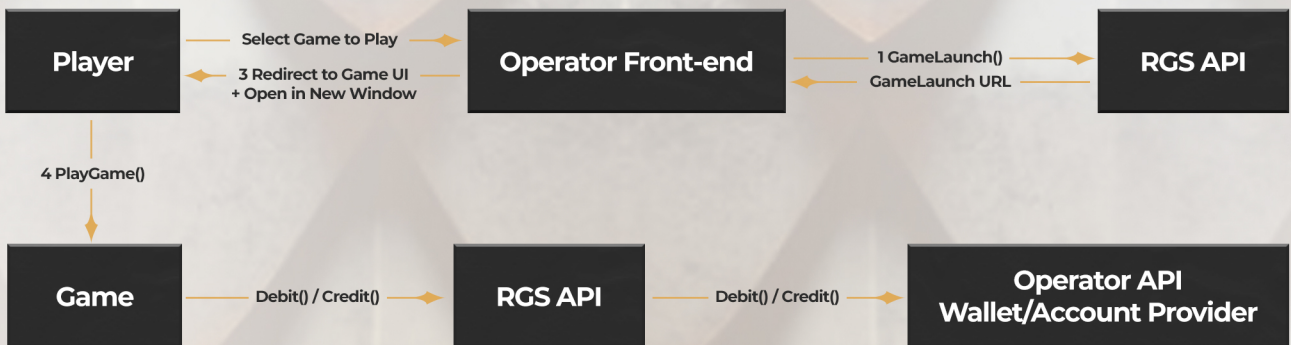
GetGames() Flow



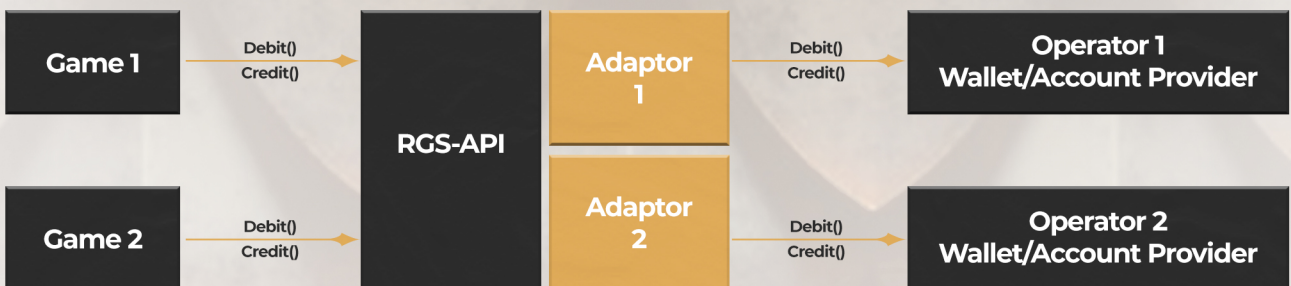
GetLaunch() Flow



Wallet Typical Flow



Debit() & Credit() Flow



GetGames() API

This API request allows the operator to receive a full list of our games which are enabled for their particular front-end. If no games are enabled, the response would return as “empty”.

GetGames() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/games/get-games/
Headers	“Content-Type”:”application/json”
POST Parameters	json object e.g.: { “api_key”: “1pghj14v5apt2bks” }

Parameters	Type	Description
api_key	String	API Ket allocation to each frontend

GetGames Request Sample

```
{  
  “api_key”: “1pghj14v5apt2bks”  
}
```



Get Games Response Example

Response would be in JSON format. Type: dictionary. Dictionary contains a root level key result and game details as value. Data in the response is self-explanatory.

```
{
  "result": {
    "games": {
      "slots": [
        {
          "game-fruityfeast": {
            "game_id": 4,
            "game_name": "fruityfeast",
            "game_title": "Fruity Feast",
            "category": "slots",
            "supplier": "test_games",
            "story": "Reap the Fruity Rewards",
            "logos": [
              {
                "url": "https://test-gam-
ing.com/images/lobby/200x150/fruityfeast.png",
                "width": "200",
                "height": "150"
              },
              {
                "url": "https://test-gam-
ing.com/images/lobby/400x300/fruityfeast.png",
                "width": "400",
                "height": "300"
              }
            ],
            "launch_params": [
              {
                "width": 800,
                "height": 600,
                "resizable": true,
                "scrollbars": false,
                "type": "browser",
                "launch_type": "new_window",
                "window_title": "%full_site_code%_%category%",
                "channel": "desktop",
                "launch_url": "https://test-
games.testsite.com/game_launcher.php?
session_id=%session_id%&channel=desktop&full_site_code=PFMNGOP&language=en&
game_name=%game_id%&category=slots&amount_
type=%amount_type%&reality_check=%reality_check%"
              },
              {
                "type": "browser",
                "launch_type": "same_window",
                "channel": "mobile",
                "launch_url": "https://test-
games.testsite.com/game_launcher.php?
session_id=%session_id%&channel=mobile&
full_site_code=PFMNGOP&language=en&game_name=%game_id%&
category=slots&amount_type=%amount_type%&reality_check=%reality_check%"
              }
            ],
            "amount_types": [
              {
                "id": "cash",
                "value": 1,
                "name": "Play"
              },
              {
                "id": "bonus",
                "value": "2",
                "name": "Play"
              }
            ]
          }
        },
        {
          "game2": "game_details"
        }
      ],
      "table_games": [],
      "scratch_cards": []
    }
  }
}
```



Parameters	Type	Description
result	JSON Object	Key values would be displayed as a dictionary.
games	JSON Object	Presented as a key inside the dictionary.
game_type	JSON Object	Inside each game is a list containing game specific details in dictionary form.

GetGames() Error Response

Any response would be provided in JSON format. Display Type: dictionary. The dictionary contains two keys names "error" and "error_details". If an error occurs, the error key will contain the value 1 and the error_details key would contain the information detailing that specific error. Any displayed messages would be setup as appropriate to each error.

```
{
  "api_key": "1"
  "error_detail": {
    "id": "1001",
    "code": "INVALID_API_KEY",
    "message": "Invalid API key."
  }
}
```

EnterGame() API .

Operator needs to make EnterGame() API request to RGS API which allows the player to enter the game. RGS API validates player's credentials, creates and registers session ID (token) at **DG back-end**, builds the launch URL and sends it back to the caller.

EnterGame() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/games/enter-game/
Headers	"Content-Type": "application/json"
POST Parameters	<pre>json object e.g.: { "api_key": "1pghj14v5apt2bks", "provider": "dragongaming", "game_type": "slots", "game_id": 1032, "platform": "desktop", "language": "en", "amount_type": "real", "username": "test_username", "password": "passt84fas28", }</pre>



Parameters	Type	Description
api_key	String	API Key allocated to each front-end
provider	String	Game Provider. DragonGaming™ here.
game_type	String	Type of Game (eg: slots, table_games, scratch_cards)
game_id	Integer	Game ID provided by the provider
platform	String	Channel in which player is playing the game. Ex: desktop, mobile
language	String	Language the game will be played in.
amount_type	String	Mode of Play For Example: 'real' – when playing with real cash 'fun' – when playing fun mode 'promo_freespin' – when playing with promotions.
username	String	Player's username. Note: in 'fun' mode arbitrary username should be used, e.g. "username": "fun"
password	String	Player's password. Note: in 'fun' mode arbitrary password should be used, e.g. "password": "fun"

EnterGame() Response Example

Response would be in JSON format. Type: *dictionary*.

```
{
  "result": {
    "launch_url": "https://staging-games.dragongaming.com/game_launcher.php?
      session_id=b6ef88a8054e328f4459b625baf38fae71981a34&
      channel=desktop&full_site_code=PFMNGOP&
      language=en&game_name=kingofgods&category=slots&
      amount_type=1&reality_check=120"
  }
}
```



Object / Key in response object	Type	Description
launch_url	JSON Object	Key launch_url contains the Game Launch URL as value that is used to launch the game.

GameExit() API.

Operator makes ExitGame() API request to RGS API to exit the player from the game. RGS API deletes the session ID of the player that was created earlier in **DG back-end**.

GameExit() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/games/game-exit/
Headers	"Content-Type": "application/json"
POST Parameters	<pre>json object e.g.: { "api_key": "1pghj14v5apt2bks", "username": "test_username", }</pre>

Parameters	Type	Description
api_key	String	API allocated to each front-end
username	String	Player's username

GameExit() Response Example

Response would be in JSON format. Type: *dictionary*.

```
{
  "result": {
    "status": "success"
  }
}
```



Object / Key in response object	Type	Description
result	JSON Object	Key result contains the JSON object as value.
status	String	Response status

CreatePlayer() API.

This API request enables the operator to register a player in DG system. Password sent in POST parameters by the operator must be encrypted.

CreatePlayer() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/v1/player/create-player/
Headers	"Content-Type": "application/json"
POST Parameters	<pre> json object e.g.: { "api_key": "1pghj14v5apt2bks", "player_id": "1025", "username": "test_username", "password": "passt84fas28", "currency": "GBP", "country": "HK", } </pre>

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
player_id	String	Player ID of the player known to the operator
username	String	Username of the player
password	String	Player's password
currency	String	3 letter ISO currency symbol
country	String	2 letter ISO country code



CreatePlayer() Response Example

Response would be in JSON format. Type: *dictionary*.

```
{
  "result": {
    "status": "success"
  }
}
```

Object / Key in response object	Type	Description
result	JSON Object	Key result contains the JSON object as value.
status	String	Response status.

UpdatePassword() API .

This API request enables the operator to update the password of the player in DG system.

UpdatePassword() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/player/update-password/
Headers	"Content-Type": "application/json"
POST Parameters	json object e.g.: { "api_key": "1pghj14v5apt2bks", "username": "test_username", "password": "passt84fas28", "new_password": "newPassw0rd", }

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
username	String	Username of the player
password	String	Player's current password
new_password	String	Player's new password

UpdatePassword() Response Example

Response would be in JSON format. Type: *dictionary*.

```
{  
  "result": {  
    "status": "success"  
  }  
}
```

Object / Key in response object	Type	Description
result	JSON Object	Key result contains the JSON object as value.
status	String	Response status



WalletDeposit() API .

This API request enables the operator to deposit funds from operator's main wallet to provider's game wallet.

WalletDeposit() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/wallet/wallet-deposit/
Headers	"Content-Type": "application/json"
POST Parameters	json object e.g.: { "api_key": "1pghj14v5apt2bks", "username": "test_username", "password": "passt84fas28", "amount": 200, "transaction_id": "10032" }

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
username	String	Username of the player
password	String	Password of the player
amount	Decimal	Amount to be deposited into DG game wallet
transaction_id	String	Transaction ID
bonus_id	String	Provider will send the bonus / campaign ID which is related to the promotional freespins. It will be null when it is in real mode.

WalletDeposit() Response Example

Response would be in JSON format. Type: *dictionary*.

```
{  
  "result": {  
    "status": "success",  
    "amount": 200.00,  
    "currency": "EUR"  
  }  
}
```



Object / Key in response object	Type	Description
result	JSON Object	Key result contains the JSON object as value.
status	String	Response status
amount	Decimal	Player's balance after deposit
currency	String	Player's currency.

WalletWithdraw() API.

This API request enables the operator to get the remaining funds from provider's game wallet to operator's main wallet. It updates the game wallet by deducting the requested withdrawal amount.

WalletWithdraw() request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/wallet/wallet-withdraw/
Headers	"Content-Type": "application/json"
POST Parameters	<pre>json object e.g.: { "api_key": "1pghj14v5apt2bks", "username": "test_username", "password": "passt84fas28", "amount": 200, "transaction_id": "10032" }</pre>

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
username	String	Username of the player
password	String	Password of the player
amount	Decimal	Amount to be deposited into DG game wallet
transaction_id	String	Transaction ID

WalletWithdraw() Response Example

Response would be in JSON format. Type: *dictionary*.

```
{
  "result": {
    "status": "success",
    "amount": 200,
    "currency": "GBP"
  }
}
```



Object / Key in response object	Type	Description
result	JSON Object	Key result contains the JSON object as value.
status	String	Response status
amount	decimal	Player's balance after withdraw
currency	String	3 letter ISO currency code

WithdrawStatus() API.

This API request enables the operator to grant bonus free spins (promotional free spins) to the player.

WithdrawStatus() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/wallet/wallet-withdraw-status/
Headers	"Content-Type": "application/json"
POST Parameters	<pre> json object e.g.: { "api_key": "1pghj14v5apt2bks", "username": "test_username", "password": "passt84fas28", "transaction_id": "10032" } </pre>

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
username	String	Username of the player
password	String	Password of the player
transaction_id	String	Unique transaction ID

WithdrawStatus() Response Example

Response would be in JSON format. Type: *dictionary*.

```

{
  "result": {
    "status": "success",
    "transaction_id": "10032",
  }
}

```



Object / Key in response object	Type	Description
result	JSON Object	Key result contains the JSON object as value.
status	String	Response status
transaction_id	String	Transaction ID

WalletTransactionStatus() API.

This API request enables the operator to get the status of wallet transaction. Transaction can be either deposit or withdraw by operator.

WalletTransactionStatus() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/wallet/wallet-transaction-status/
Headers	"Content-Type": "application/json"
POST Parameters	<pre> json object e.g.: { "api_key": "1pghj14v5apt2bks", "username": "test_username", "password": "passt84fas28", "transaction_type": "deposit", "transaction_id": "10032" } </pre>

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
username	String	Username of the player
password	String	Password of the player
transaction_type	String	Type of transaction. It can be either 'deposit' or 'withdraw'
transaction_id	String	Unique transaction ID

WalletTransactionStatus() response example

```

{
  "result": {
    "status": "success",
    "transaction_id": "10032",
  }
}

```



Object / Key in response object	Type	Description
result	JSON Object	Key result contains the JSON object as value.
status	String	Response status
transaction_id	String	Transaction ID

WalletGetBalance() API.

This API request enables the operator to get the available balance of a player.

WalletGetBalance() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/wallet/wallet-get-balance/
Headers	"Content-Type": "application/json"
POST Parameters	<pre> json object e.g.: { "api_key": "3c2ef39d02581e2db3ddf6c713e83e05", "username": "123123", "password": "1234yF" } </pre>

Parameters	Type	Description
api_key	String	API Key allocated to each frontend
username	String	Username of the player
password	String	Player's password

WalletGetBalance() Response Example

```

{
  "status": "success",
  "amount": 12345,
  "currency": "USD"
  "account_id": "123123"
}

```



Object / Key in response object	Type	Description
Status	Integer	Response status
Amount	Decimal	Latest balance of the player
Currency	String	Currency of the player
account_id	String	Player's unique ID

PlayerBets() API.

This API request enables the operator to get summarised betting activity for a player in a given period.

PlayerBets() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/player/player-bets/
Headers	"Content-Type": "application/json"
POST Parameters	<pre> json object e.g.: { "api_key": "1pghj14v5apt2bks", "username": "test_username", "start_date": "2020-02-20", "end_date": "2020-02-21", "amount_type": "real", } </pre>

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
username	String	Username of the player
start_date	String	Start date for period of interest
end_date	String	End date for period of interest
amount_type	String	Type of play



PlayerBets() response example

```
{
  "result": {
    "status": "success",
    "username": "test_username",
    "account_id": "10032",
    "num_bets": 50,
    "total_bet_amount": 1050,
    "total_win_amount": 920,
    "currency": "GBP"
  }
}
```

Object / Key in response object	Type	Description
result	JSON Object	Key result contains the JSON object as value.
status	String	Response status
username	String	Username of the player
account_id	String	Account ID of the player
num_bets	Integer	Number of bets the player has placed during the given period
total_bet_amount	Integer	Total amount of money the player has bet during the given period
total_win_amount	Integer	Total amount of money the player has won during the given period
currency	String	3 letter ISO currency code

GameHistory() API .

This API request enables the operator to get history of recent 100 game rounds played with real cash mode by default.

GameHistory() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/games/game-history/
Headers	"Content-Type": "application/json"
POST Parameters	json object e.g.: { "api_key": "1pghj14v5apt2bks", "player_id": "1002", "amount_type": "real", "start_date": "2020-02-20 00:00:00", "end_date": "2020-02-21 00:00:00", "page_num": 1, }

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
player_id	String	ID of the player
amount_type	String	Mode using which game was played
start_date	String	End date time in the following format "YYYY-MM-DD HH:MM:SS"
end_date	String	End date time in the following format "YYYY-MM-DD HH:MM:SS"
page_num	Integer	Page number

GameHistory() Response Example

Response would be in JSON format. Type: dictionary.

```
{
  "game_history": {
    "headers": [
      "Game Name",
      "Round ID",
      "Bet Amount", "Win
      Amount", "Amount Type", "Date Time"],
    "data": [
      [
        "Test Game Name-1",
        1001,
        100,
        100,
        "real",
        "2019-07-23 17:20:57"
      ],
      [
        "Test Game Name-1",
        1002,
        100,
        100,
        "real",
        "2019-07-23 17:20:57"
      ],
      [
        "Test Game Name-2",
        1003,
        100,
        100,
        "real",
        "2019-07-23 17:20:57"
      ]
    ]
  }
}
```

Object / Key in response object	Type	Description
game_history	JSON Object	Response status
game_history["headers"]	JSON List	Table headers
game_history["data"]	JSON List of Lists	List of Lists of game rounds data

GameHistoryAllPlayers() API .

This API request enables the operator to get the game history of all the players for a given time period. For now, RGS API sends up to 30 minutes of data per request.

GameHistoryAllPlayers() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/games/games-history-all-players/
Headers	"Content-Type": "application/json"
POST Parameters	<pre>json object e.g.: { "api_key": "1pghj14v5apt2bks", "amount_type": "real", "start_date": "2020-05-26 13:30:00", "end_date": "2020-05-26 14:00:35", }</pre>

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
amount_type	String	Mode using which game was played
start_date	String	Start date time in the following format "YYYY-MM-DD HH:MM:SS"
end_date	String	End date time in the following format "YYYY-MM-DD HH:MM:SS"

GameHistoryAllPlayers() Response Example

Response would be in JSON format. Type: dictionary.

```
{
  "game_history": {
    "headers": [
      "Player ID",
      "Game ID",
      "Game Name",
      "Game Category",
      "Platform",
      "Round ID",
      "Bet Amount",
      "Win Amount",
      "Amount Type",
      "Play Type",
      "Date Time",
      "Bet Status"
    ],
    "data": [
      [
        "1001",
        "1001",
        "Game-1 Title",
        "slots",
        "desktop",
        1234,
        100,
        0,
        "real",
        "normal",
        "2020-05-05 10:20:57",
        "lose"
      ],
      [
        "1001",
        "1001",
        "Game-2 Title",
        "slots",
        "desktop",
        1234,
        100,
        10,
        "real",
        "freespins",
        "2020-05-05 10:20:57"
      ],
      [
        "1001",
        "1001",
        "Game-3 Title",
        "slots",
        "desktop",
        1234,
        100,
        20,
        "real",
        "normal",
        "2020-05-05 10:20:57",
        "lose"
      ],
      [
        "1001",
        "1001",
        "Game-4 Title",
        "slots",
        "desktop",
        1234,
        100,
        0,
        "real",
        "respin",
        "2020-05-05 10:20:57",
        "win"
      ]
    ]
  }
}
```


Object / Key in response object	Type	Description
game_history	JSON Object	Response status
game_history['headers']	JSON List	Table headers
game_history['data']	JSON List of Lists	List of Lists of game rounds data

GrantBonus() API .

This API request enables the operator to grant bonus free spins (promotional free spins) to the player.

GrantBonus() Request

Type	Input
HTTP Method	POST
API URL	https://<API_BASE_URL>/wallet/grant-bonus/
Headers	"Content-Type": "application/json"
POST Parameters	<pre> json object e.g.: { "api_key": "150Y0Rx1BC0hYbCm", "amount_type": "promo_freespin", "campaign_id": "B0n-3", "game_ids": [1032, 1041], "coin_value_level": 1, "num_rounds": 10, "player_ids": ["267"], "currencies": ["USD"], "start_date": "2021-01-19 14:00:00", "end_date": "2021-01-30 15:00:00", } </pre>

Parameters	Type	Description
api_key	String	API Key allocated to each front-end
amount_type	String	Mode of the play. Ex: 'promo_freespin' – It is when the player is playing with promo freespins
campaign_id	String	It should be unique campaign id
game_ids	List	List of game_id for which promo free spins are granted. Ex: game_ids: [1,10,6,3]
coin_value_level	Integer	The value must be an integer and must be from allowed list of coin values levels. Before you start with promo freespins, we will provide the list of allowed coin value levels. Ex: coin_values:[10, 10, 20, 30]
num_rounds	Integer	It indicates the number of free spin rounds that is granted to the player.
player_ids	List	Player ID of the player known to the operator. Free spins are granted to all these player IDs mentioned in this list and for all the games mentioned in the game_ids list.
currencies	List	Currency of the player and it should match with the player currency which is already registered with us. Currencies count should match with the players count. Ex:- ["EUR", "USD"]
start_date	String(Optional)	Date time from which this bonus can be used. It should not be less than are equal to current time format = "YYYY-MM-DD HH:MM:SS"
end_date	String	Expire date time of the promotion. It cannot be less than the start_date format = "YYYY-MM-DD HH:MM:SS"
max_win_limit (optional)	Integer	This parameter is used to cap the winnings from promo free spins. We need to provide this value while granting free spins. It's value is in cents. It should be 0 when there is no cap on the winnings. For example, Player with USD currency is awarded with 10 free spins with 100 cents as cap amount. If the winnings are USD 20 from these 10 rounds, as per the cap amount, only 1 USD is credited to the player wallet. If the cap value is set a 0, entire win amount of USD 20 is credited to player wallet.

GrantBonus() API Additional Notes

Combination of campaign_id, player_id and game_id is always unique per operator/aggregator. Coin value levels are explained below. Below are the coin values per game and per currency for reference. Actual coin values will be provided during the integration.

Country	Currencies	Level 1	Level 2	Level 3	Level 4	Level 5
European Union	EUR	1	2	3	4	5
Argentina	ARS	30	60	90	120	150
Australia	AU	1	2	3	4	5
Brazil	BRL	5	10	15	20	25
Bulgaria	BGN	1	2	3	4	5
Canada	CAD	1	2	3	4	5
China	CNY	10	20	30	40	50

✦ Example: For Game-1 if the coin value level 3 is chosen, coin value for EUR player would be 3, coin value for ARS player would be 90, coin value for BRL would be 15. These are the coin values that will be used during the promo free spins play.

GrantBonus() Response & Example

```
{
  "result": {
    "status": "success",
    "promo_freespin_id": 131
  }
}
```

Object / Key in response object	Type	Description
result	JSON Object	Key result contains the JSON object as value.
status	String	Response status
promo_freespins_id	Integer	This is providers ID.