



## GameArt Slot Integration v3.8

June 2

2016

Communication between the 3rd party system and the GameArt Slot Service API will be through a secure HTTPS connection, where the request will be filtered by the remote servers IP. In order to begin a transaction a valid username and password is required. Please contact GameArt customer support to obtain your username and password and specify the IP of the server connection to the service API interface.

GameArt Slot Service API documentation provides a means for external systems to manage and retrieve the player or casino data from the Slot Game Engine.

The information contained in this document represents the current view of GameArt Enterprise on the issues discussed as-of the date of publication. Because GameArt Enterprise must respond to changing market conditions, it should not be interpreted to be a commitment on the part of GameArt Enterprise, and GameArt Enterprise cannot guarantee the accuracy of any information presented after the date of publication.

<b>1. Introduction</b>	<b>3</b>
<b>2. Overview</b>	<b>3</b>
<b>3. Technical overview</b>	<b>3</b>
3.1 Request and url format	3
3.2 Response	4
3.3 Common reporting methods/actions	4
<b>4. Seamless wallet</b>	<b>5</b>
4.1 Communication overview	5
4.2 Request validation	5
4.3 External system API service	5
<i>Player balance</i>	5
<i>Credit players balance</i>	6
<i>Debit players balance</i>	6
<i>External service JSON response</i>	7
<i>Status codes</i>	7
<b>5. Admin service</b>	<b>8</b>
5.1 Lists available games (action: available_games)	8
5.2 Get game url (action: get_game)	8
5.3 Get game url, play for fun (action: get_demo)	8
5.4 Change / Set casino default settings (action: settings)	9
5.5 Check all players statuses (action: players)	9
5.6 Check if player exists (action: check_player)	9
5.7 Deposit funds to player account (action: deposit)	10
5.8 Withdraw funds from player account (action: withdraw)	10
5.9 Withdraw all funds from player account (action: withdraw_all)	10
5.10 Refund funds to player account (action: refund)	11
5.11 Get player balance (action: player_balance)	11
5.12 Create player (action: create_player)	11
5.13 Set or update player status (action: player_status)	12
5.14 Player logout, deletes all active game sessions (action: player_logout)	12
5.15 Ping server (action: ping)	12
5.16 Get game round info (action: game_round)	13
5.16.1 Response explained	13
<b>6. Freerounds service</b>	<b>14</b>
6.1 Get game bets (action: bets)	14
6.2 Add freerounds (action: add)	14
6.3 Show current freerounds (action: show)	15
6.4 Remove freerounds (action: remove)	15

## 1. Introduction

Slot Game Engine is a game provider created by GameArt Enterprise.

This document describes Customer side of integration. Customer integration consists from integration of Slot Game Engine games into Customer platform for players of Customer.

## 2. Overview

Slot Service API offers three service sections divided by usage. Admin api is used for handling casino and player related actions:

SEAMLESS & WALLET:

- Request game url (real/fun), create game sessions
- Request available games
- Request reports
- Change casino settings
- Award Freerounds

WALLET ONLY:

- Create player, logout player
- Deposit, withdraw, refund player funds
- Check players and player balance

Report api is used for generating game and transactions (deposit/withdraw) reports within selected date range and last but not least, the game api is used by the game user interface.

## 3. Technical overview

All service sections follow a common set of rules

### 3.1 Request and url format

The request will be a „GET“ url request containing following parameters used in query string and content data.

Base url:

- `https://<url>/!v8/api/admin.js` - Admin service
- `https://<url>/!v8/api/report.js` - Report service

Parameters:

- `action` – name of method trying to execute
- `args array[0][var]` – method's arguments as array
- `args array[1][var]` – authentication arguments (usr/passw)

### 3.2 Response

Response is always an array with associative keys, “status” which is code that equals HTTP status codes and “response” which is the response of the method called (a.k.a. action). For demonstrational purposes all further responses will be rendered in JSON (recommended data type)

Example: {“status”: 200, “response”: true}

### 3.3 Common reporting methods/actions

Reporting sections share a common action “analytics”, which can be used with different “report” arguments:

args[0][report]	additional args	description
<b>player</b>		Show all players and their info
player	remote_id	Show info for specific player
player	remote_id, game_id	Show info for specific player per game
player	game_id	Show all players and their info per game
<b>game</b>		Show all games and their info
game	remote_id	Show all game info for specific player
game	game_id	Show info for specific game
<b>transactions</b>		Show all transactions (withdraws and deposits)
transactions	remote_id	Show all transactions for specific player
<b>history</b>	game_id, session_id, remote_id	Show all game rounds and filter by nothing, game_id, session_id, remote_id or by any combination of the additional arguments
important args		description
	start	Start date of desired timespan (2015-01-31 10:59:59)
	end	End date of desired timespan (2015-05-31 04:04:04)
	max	Specify maximum number of outputs

**NOTE:** Important arguments to use in reporting are “start” and “end” as they determine the time-span of the requested report. If you do not specify “start”, only last 24h will be reported. If no “end” is specified then end time is current time.

**Request:** `https://<url>/lv8/api/report.js?action=analytics&args[1][usr]=username&args[1][passw]=password&args[0][report]=player&args[0][remote_id]=123&args[0][game_id]=11&args[0][start]=2015-05-28 10:17:44&args[0][end]=2015-05-29 10:17:44&args[0][max]=1`

**Response:** {“status”: 200, “response”: {“credit”: 4.7, “credit\_fs”: 0, “debit”: 15.45, “debit\_mobile”: 0, “debit\_fs”:0, “debit\_fs\_mobile”: 0, “plays”: 29, “wins”: 8}}

## 4. Seamless wallet

Seamless wallet allows players to place bets directly to external system without requiring to transfer funds between accounts.

### 4.1 Communication overview

Communication between servers go through secure https RESTful service. Service must be implemented by external system according to specification. Game Engine sends GET requests to remote system service and expects a JSON response with a status code and players balance after transaction (Example: {"status": "200", "balance": "300"}). Status code equals HTTP status codes and only 200 (OK) is accepted as a successful.

### 4.2 Request validation

Each request is escorted with a query param key with a value of a sha1 cached query string and a salt key provided by GameArt.

**Example:**

*action=balance&remote\_id=123&session\_id=123-abc&key=38432ff064690c9b03da519d0c685b1045451c9e*

Query param key value must match *sha1([SALT KEY]+[QUERY STRING])*.

**QUERY STRING example:** *action=balance&remote\_id=123&session\_id=123-abc*

### 4.3 External system API service

External system must provide a secure https url that responds to the following GET requests with a JSON response with a status code and players balance after transaction.

#### Player balance

Returns players current balance.

Query string name	type (max)	description	mandatory Y/N
action	string	External service action [ <b>balance</b> , credit, debit]	Y
remote_id	varchar	Unique player id	Y
session_id	alphanumeric(32-40)	Unique game session id generated by provider	N
remote_data	any	Remote data provided by the service caller (casino) for remote session conformation or any data that needs to be passed from casino through our session	N
key	string(40)	Request validation cache key	Y

**Request:** <REMOTE

URL>/?action=balance&remote\_id=1&session\_id=123456789012345678901324567890abcd&remote\_data={"useful\_id": "12dar67890123iemk  
sueskt4567980abcd"}&key=38432ff064690c9b03da519d0c685b1045451c9e

**Response:** {"status": "200", "balance": "300.00"}

## Credit players balance

Returns player balance after credit transaction. Credit is called after every debit, even if credit amount is 0, this indicates that a game round is complete.

There are 3 types of credit actions that should be handled by the customer:

- WIN should credit the amounted funds to the player,
- REF is a refund action that is called from GameArt server to the customer and indicates a refund of credit to the player. It only sends mandatory parameters (no game\_id, round\_id, session\_id, remote\_data),
- WIN\_FREE indicates a win from freerounds that should also be normally credited (only used with freerounds feature explained in chapter 6)

Query string name	type (max)	description	mandatory Y/N
action	string	External service action [balance, <b>credit</b> , debit]	Y
action_type	string	Type of <b>credit</b> action [ <b>win</b> , <b>win_free</b> (check chapter 6)]	Y
amount	double(10,2)	Amount to credit players account	Y
remote_id	varchar	Unique player id	Y
transaction_id	integer	Unique transaction id for current transaction	Y
game_id	integer(3)	Unique game id, see action available_games	N
round_id	integer	Game round id for current running game round	N
session_id	alphanumeric(32-40)	Unique game session id generated by provider	N
remote_data	any	Remote data provided by the service caller (casino) for remote session conformation or any data that needs to be passed from casino through our session	N
key	string(40)	Request validation cache key	Y

**Request:** <REMOTE

URL>/?action=credit&action\_type=WIN&amount=0.3&remote\_id=1&transaction\_id=27&game\_id=3&round\_id=123&session\_id=123456789012345678901324567980abcd&remote\_data={"useful\_id":"12dar67890123iemksueskt4567980abcd"}&key=49f749364b129d9f91d2bef7dd044a93af0fb676

**Response:** {"status":200,"balance":"300.00"}

## Debit players balance

Returns player balance after debit transaction.

Query string name	type (max)	description	mandatory Y/N
action	string	External service action [balance, <b>credit</b> , debit]	Y
action_type	string	Type of <b>debit</b> action [ <b>bet</b> , <b>bet_free</b> (check chapter 6)]	Y
amount	double(10,2)	Amount to credit players account	Y
remote_id	varchar	Unique player id	Y
transaction_id	integer	Unique transaction id for current transaction	Y
game_id	integer(3)	Unique game id, see action available_games	N
round_id	integer	Game round id for current running game round	N
session_id	alphanumeric(32-40)	Unique game session id generated by provider	N
remote_data	any	Remote data provided by the service caller (casino) for remote session conformation or any data that needs to be passed from casino through our session	N
key	string(40)	Request validation cache key	Y

**Request:** <REMOTE

URL>/?action=debit&action\_type=BET&amount=0.3&remote\_id=1&transaction\_id=27&game\_id=3&round\_id=123&session\_id=123456789012345678901324567980abcd&remote\_data={"useful\_id":"12dar67890123iemksueskt4567980abcd"}&key=49f749364b129d9f91d2bef7dd044a93af0fb676

**Response:** {"status":200,"balance":"300.00"}

## External service JSON response

JSON key	type(max)	Description	mandatory Y/N
status	integer(3)	HTTP status code	Y
balance	double(10,2)	Players balance (after transaction)	Y
msg	string(255)	Error message	N

**Example:** {"status":200,"balance":"300.00"}, {"status":403,"balance":"0.30","msg":"Insufficient funds"}, {"status":500,"msg":"Black magic"}

## Status codes

External service can respond with 3 type of status codes. 200 for success, 500 service error and 403 for debit transaction refusal. Only status of 200 will be accepted as OK by the GameArt server.

## 5. Admin service

Request base URL: <https://<url>/lv8/api/admin.js>

Most of admin services are not needed for **seamless** integration but **wallet** integration does use many of them. Important services that are needed for seamless integration are: **available\_games**, **get\_game**, **get\_demo** and **game\_round** (for reporting).

### 5.1 Lists available games (action: available\_games)

args (args[0])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:** [https://<url>/lv8/api/admin.js?action=available\\_games&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password](https://<url>/lv8/api/admin.js?action=available_games&args[1][usr]=username&args[1][passw]=password)

**Response:** { "status": 200, "response": [ { "id": "8", "name": "Dragon King", "alias": "DragonKing", "brand": "gameart", "rtp": [ 94, 96 ], "lines": 50 }, { "id": "9", "name": "Wild Dolphin", "alias": "WildDolphin", "brand": "gameart", "lines": 10, "rtp": [ 94, 96 ] }, { "id": "12", "name": "Venetia", "alias": "Venetia", "brand": "gameart", "rtp": [ 96 ], "lines": 20 } ] }

### 5.2 Get game url (action: get\_game)

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y
game_id	integer(3)	Unique game id, see action available_games	Y
remote_data	any	Remote data provided by the service caller (casino) for remote session conformation or any data that needs to be passed from casino through our session	N

  

args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:**

[https://<url>/lv8/api/admin.js?action=get\\_game&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123&args\[0\]\[game\\_id\]=12](https://<url>/lv8/api/admin.js?action=get_game&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123&args[0][game_id]=12) (optional: &args[0][remote\_data]={ "useful\_id": "12dar67890123iemksueskt4567980abcd" })

**Response:** { "status": 200, "response": { "game\_url": "https://<url>/lv8/views/gameart/game.js?sid=abcd1-234", "token": "abcd1-234", "config": { "scale": "small", "betMultipliers": [ 1, 2, 5, 10, 15, 25 ], "initialBetMultiplierIndex": 2, "game\_api": "https://<url>/lv8/gameserver.js?sid=abcd1-234", "game\_base": "" } } }

### 5.3 Get game url, play for fun (action: get\_demo)

args (args[0])	type (max)	description	mandatory Y/N
game_id	integer(3)	Unique game id, see action available_games	Y

  

args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:** [https://<url>/lv8/api/admin.js?action=get\\_game&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[game\\_id\]=12](https://<url>/lv8/api/admin.js?action=get_game&args[1][usr]=username&args[1][passw]=password&args[0][game_id]=12)

**Response:** { "status": 200, "response": { "game\_url": "https://<url>/lv8/views/gameart/game.js?sid=abcd1-234", "token": "abcd1-234", "config": { "scale": "small", "betMultipliers": [ 1, 2, 5, 10, 15, 25 ], "initialBetMultiplierIndex": 2, "game\_api": "https://<url>/lv8/gameserver.js?sid=abcd1-234", "game\_base": "" } } }



## 5.4 Change / Set casino default settings (action: settings)

args (args[0])	type (max)	description	mandatory Y/N
render	string	Default rendering type for the casino. Must be one of the following options ('json', 'xml', 'plain-text').	N
session_expire	integer(10)	Casino session expire time in seconds (default: 30 sec)	N
session_start_expire	integer(10)	How many seconds does the client have to load/make first request before session is cleared (default: 120 sec)	N
media	string(50)	Casinos media server (default: localhost)	N
service	string(50)	Casinos service server (default: localhost)	N
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**NOTE:** If no argument is provided, the current casino settings are displayed.

**Request:** [https://<url>/lv8/api/admin.js?action=settings&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[service\]=newserviceurl](https://<url>/lv8/api/admin.js?action=settings&args[1][usr]=username&args[1][passw]=password&args[0][service]=newserviceurl)

**Response:** {"status": 200, "response": true}

## 5.5 Check all players statuses (action: players)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:** [https://<url>/lv8/api/admin.js?action=players&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password](https://<url>/lv8/api/admin.js?action=players&args[1][usr]=username&args[1][passw]=password)

**Response:** {"status": 200, "response": [ { "remote\_id": 123, "status": "ACTIVE" }, { "remote\_id": "321", "status": "ACTIVE" }, { "remote\_id": 213, "status": "ACTIVE" }, { "remote\_id": 312, "status": "ACTIVE" } ] }

## 5.6 Check if player exists (action: check\_player)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:** [https://<url>/lv8/api/admin.js?action=check\\_player&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123](https://<url>/lv8/api/admin.js?action=check_player&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123)

**Response:** {"status": 200, "response": true}}

## 5.7 Deposit funds to player account (action: deposit)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y
amount	integer	Amount to deposit	Y
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:**

[https://<url>/lv8/api/admin.js?action=deposit&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123&args\[0\]\[amount\]=12123](https://<url>/lv8/api/admin.js?action=deposit&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123&args[0][amount]=12123)

**Response:** {"status": 200, "response": "1212300"}

## 5.8 Withdraw funds from player account (action: withdraw)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y
amount	integer	Amount to withdraw	Y
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:**

[https://<url>/lv8/api/admin.js?action=withdraw&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123&args\[0\]\[amount\]=12123](https://<url>/lv8/api/admin.js?action=withdraw&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123&args[0][amount]=12123)

**Response:** {"status": 200, "response": "0"}

## 5.9 Withdraw all funds from player account (action: withdraw\_all)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:** [https://<url>/lv8/api/admin.js?action=withdraw\\_all&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123](https://<url>/lv8/api/admin.js?action=withdraw_all&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123)

**Response:** {"status": 200, "response": "0"}

## 5.10 Refund funds to player account (action: refund)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y
amount	integer	Amount to withdraw	Y
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:**

[https://<url>/lv8/api/admin.js?action=refund&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123&args\[0\]\[amount\]=12123](https://<url>/lv8/api/admin.js?action=refund&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123&args[0][amount]=12123)

**Response:** {"status": 200, "response": "1212300"}

## 5.11 Get player balance (action: player\_balance)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y
args (args[1])	type(max)	description	mandatory Y/N
Usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:** [https://<url>/lv8/api/admin.js?action=player\\_balance&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123](https://<url>/lv8/api/admin.js?action=player_balance&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123)

**Response:** {"status": 200, "response": "0"}

## 5.12 Create player (action: create\_player)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y
fname	string(50)	Players first name	N
lname	string(50)	Players last name	N
nickname	string(50)	Players last nickname	N
pool_id	intiger(10)	Players pool id	N
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:**

[https://<url>/lv8/api/admin.js?action=create\\_player&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123&args\[0\]\[fname\]=newplayer](https://<url>/lv8/api/admin.js?action=create_player&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123&args[0][fname]=newplayer)

**Response:** {"status": 200, "response": true}

### 5.13 Set or update player status (action: player\_status)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y
status	string	One of the available player statuses ('ACTIVE', 'BLOCKED', 'PENDING')	Y

  

args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:**

[https://<url>/lv8/api/admin.js?action=player\\_balance&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123&args\[0\]\[status\]=ACTIVE](https://<url>/lv8/api/admin.js?action=player_balance&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123&args[0][status]=ACTIVE)

**Response:** {"status": 200, "response": true}

### 5.14 Player logout, deletes all active game sessions (action: player\_logout)

**NOTE:** not needed for Seamless Integration

args (args[0])	type (max)	description	mandatory Y/N
remote_id	varchar	Unique player id	Y

  

args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:** [https://<url>/lv8/api/admin.js?action=player\\_logout&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[remote\\_id\]=123](https://<url>/lv8/api/admin.js?action=player_logout&args[1][usr]=username&args[1][passw]=password&args[0][remote_id]=123)

**Response:** {"status": 200, "response": true}

### 5.15 Ping server (action: ping)

args (args[0])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:** [https://<url>/lv8/api/admin.js?action=ping&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password](https://<url>/lv8/api/admin.js?action=ping&args[1][usr]=username&args[1][passw]=password)

**Response:** {"status": 200, "response": true}

## 5.16 Get game round info (action: game\_round)

args (args[0])	type (max)	description	mandatory Y/N
game_id	integer(3)	Unique game id, see action available_games	Y
gr_id	integer	Game round id of a played game	Y
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

### Request:

[https://<url>/lv8/api/admin.js?action=game\\_round&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[game\\_id\]=12&args\[0\]\[gr\\_id\]=1234567890](https://<url>/lv8/api/admin.js?action=game_round&args[1][usr]=username&args[1][passw]=password&args[0][game_id]=12&args[0][gr_id]=1234567890)

**Response:** {"status": 200, "response": {"status": 200, "response": {"events": [{"event": "bet", "context": {"total": 125, "betPerLine": 5, "paylines": [[0, 0, 0, 0, 0], [1, 1, 1, 1, 1], [2, 2, 2, 2, 2], [3, 3, 3, 3, 3], [0, 0, 1, 0, 0], [0, 1, 1, 1, 0], [0, 1, 2, 1, 0], [0, 1, 0, 1, 0], [0, 0, 0, 1, 0], [0, 1, 0, 0, 0], [0, 0, 1, 1, 0], [0, 1, 1, 0, 0], [0, 0, 2, 0, 0], [0, 0, 2, 1, 0], [0, 1, 2, 0, 0], [1, 1, 2, 1, 1], [1, 1, 0, 1, 1], [1, 2, 2, 2, 1], [1, 0, 0, 0, 1], [1, 2, 1, 2, 1], [1, 0, 1, 0, 1], [1, 1, 1, 2, 1], [1, 1, 1, 0, 1], [1, 2, 1, 1, 1], [1, 0, 1, 1, 1], [1, 2, 0, 2, 1], [1, 0, 2, 0, 1], [2, 2, 3, 2, 2], [2, 2, 1, 2, 2], [2, 3, 3, 3, 2], [2, 1, 1, 1, 2], [2, 3, 2, 3, 2], [2, 1, 2, 1, 2], [2, 2, 2, 3, 2], [2, 2, 2, 1, 2], [2, 3, 2, 2, 2], [2, 1, 2, 2, 2], [2, 3, 1, 3, 2], [2, 1, 3, 1, 2], [3, 3, 2, 3, 3], [3, 2, 2, 2, 3], [3, 2, 1, 2, 3], [3, 2, 3, 2, 3], [3, 3, 3, 2, 3], [3, 2, 3, 3, 3], [3, 3, 2, 2, 3], [3, 2, 2, 3, 3], [3, 3, 1, 3, 3], [3, 3, 2, 1, 0], [3, 2, 1, 0, 0]], "maxWinCap": 0}], {"event": "gameStart", "context": {"totalBet": 125, "betPerLine": 5}}, {"event": "spinStart", "context": {"symbols": ["WILD", "PIC1", "PIC2", "PIC3", "PIC4", "ACE", "KING", "QUEEN", "JACK", "TEN", "NINE", "SCAT"], "symbolsPay": {"line": ["WILD", "PIC1", "PIC2", "PIC3", "PIC4", "ACE", "KING", "QUEEN", "JACK", "TEN", "NINE"], "scatter": ["SCAT"]}, "wildSymbols": ["WILD"], "lineAlign": "left", "lineCoinciding": false}}, {"event": "playedSpin", "context": [{"TEN", "NINE", "KING", "QUEEN"}, ["PIC4", "QUEEN", "KING", "QUEEN"], ["PIC1", "ACE", "PIC3", "JACK"], ["PIC4", "QUEEN", "SCAT", "QUEEN"], ["PIC3", "QUEEN", "KING", "PIC2"]]}, {"event": "gameEnd", "context": {"win": 0}}, {"event": "gameRoundOver", "context": {"win": 0}}], "actions": [{"action": "bet", "context": [50, 5]}, {"action": "play", "context": null}]}]}

### 5.16.1 Response explained

There are six events in the response (**bet**, **gameStart**, **spinStart**, **playedSpin**, **gameEnd**, **gameRoundOver**) and two actions (bet, play).

#### Events:

- **bet** - contains all the paylines and starts the game round
  - total – total bet amount
  - betPerLine – bet per line
  - paylines – defines all the lines for this game round
  - maxWinCap – if there is a cap set it is written here
- **gameStart** - starts the game. With freespins there are multiple **gameStart**, **spinStart**, **playedSpin** and **gameEnd** events for each game.
  - totalBet
  - betPerLine
- **spinStart** – contains all the defined symbols
  - symbols – all the symbol names for this game
  - symbolsPay – all the symbol name that pay
    - line – all the symbols that pay lines
    - scatter – all the scatter symbols
  - wildSymbols – wild symbol name
  - lineAlign – defines from which side the lines evaluated (left, right)
  - lineCoinciding -
- **playedSpin** – contains all the symbols that are currently on the reel
- **gameEnd** – contains the end of the game winnings
  - win – winnings of one game
- **gameRoundOver** – end of the game round, which can be only one game or in the case of a freespins multiple games. Contains all the game round winnings.
  - win – winnings for the game round

## 6. Freerounds service

Request base URL: <https://<url>/lv8/api/freerounds.js>

Freerounds service enables the client (casino) to award a desired number of free bets to their customer.

GameArt system will indicate a freeround trough **debit** and **credit** requests with an appropriate action type (**&action\_type=bet\_free** or **&action\_type=win\_free**).

### 6.1 Get game bets (action: bets)

This service lists available bets for the chosen game. One of the listed bets needs to be specified by id in **add** service in order to indicate what the bet value of the given freerounds is.

args (args[0])	type (max)	description	mandatory Y/N
game_id	integer(3)	Unique game id, see action available_games	Y
remote_id	varchar	Unique player id	Y
args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

Request:

[https://<url>/lv8/api/freerounds.js?action=bets&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[game\\_id\]=12&args\[0\]\[remote\\_id\]=123](https://<url>/lv8/api/freerounds.js?action=bets&args[1][usr]=username&args[1][passw]=password&args[0][game_id]=12&args[0][remote_id]=123)

**Response:** { "status": 200, "response": { "bets": [ { "bet": 25, "lines": 50, "betMultiplier": 1, "betId": 0 }, { "bet": 50, "lines": 50, "betMultiplier": 2, "betId": 1 }, { "bet": 125, "lines": 50, "betMultiplier": 5, "betId": 2 }, { "bet": 250, "lines": 50, "betMultiplier": 10, "betId": 3 }, { "bet": 375, "lines": 50, "betMultiplier": 15, "betId": 4 }, { "bet": 625, "lines": 50, "betMultiplier": 25, "betId": 5 } ], "lines": 50, "oneCreditBuysLines": 2 } }

### 6.2 Add freerounds (action: add)

A number of freerounds are added for specific game and cover a specific bet (see chapter 6.1).

After they are added, freerounds will start being used the very next time the player places a bet or at a specific date and time if **valid\_from** is specified. They will last indefinitely until used unless **ttl** or **valid\_to** is specified.

args (args[0])	type (max)	description	mandatory Y/N
game_id	integer(3)	Unique game id, see action available_games	Y
remote_id	varchar	Unique player id	Y
bet_id	integer(3)	Unique bet id used for new freerounds (see "bet" action)	Y
count	integer(3)	Number of freerounds to be awarded	Y
ttl	integer(10)	Time to last – How much time the freerounds are valid to be used from the moment they are added. If used with valid_from, the countdown starts at the date and time specified. In seconds! (If ttl is present, valid_to is not needed)	N
valid_from	DateTime	Date and time when freerounds become valid. Can be used alone or with valid_to or with ttl. (format: 2015-01-31 10:59:59)	N
valid_to	DateTime	Date and time when freerounds expire. (format: 2015-01-31 10:59:59)	N

args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:**

[https://<url>/lv8/api/freerounds.js?action=add&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[game\\_id\]=12&args\[0\]\[remote\\_id\]=123&args\[0\]\[bet\\_id\]=3&args\[0\]\[count\]=10&args\[0\]\[valid\\_from\]=2015-01-31 10:59:59&args\[0\]\[ttl\]=20000](https://<url>/lv8/api/freerounds.js?action=add&args[1][usr]=username&args[1][passw]=password&args[0][game_id]=12&args[0][remote_id]=123&args[0][bet_id]=3&args[0][count]=10&args[0][valid_from]=2015-01-31 10:59:59&args[0][ttl]=20000)

**Response:** { "status": 200, "response": { "id": 8, "used": 0, "count": 10, "betArgs": [ 50, 10 ], "bet": { "bet": 250, "lines": 50, "betMultiplier": 10, "betId": 3 } } }

## 6.3 Show current freerounds (action: show)

args (args[0])	type (max)	description	mandatory Y/N
game_id	integer(3)	Unique game id, see action available_games	Y
remote_id	varchar	Unique player id	Y

args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:**

[https://<url>/lv8/api/freerounds.js?action=show&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[game\\_id\]=12&args\[0\]\[remote\\_id\]=123](https://<url>/lv8/api/freerounds.js?action=show&args[1][usr]=username&args[1][passw]=password&args[0][game_id]=12&args[0][remote_id]=123)

**Response:** { "status": 200, "response": [ { "id": 8, "used": 0, "count": 10, "betArgs": [ 50, 10 ], "bet": { "bet": 250, "lines": 50, "betMultiplier": 10, "betId": 3 } }, { "id": 9, "used": 0, "count": 6, "betArgs": [ 50, 10 ], "bet": { "bet": 250, "lines": 50, "betMultiplier": 10, "betId": 3 } } ] }

## 6.4 Remove freerounds (action: remove)

args (args[0])	type (max)	description	mandatory Y/N
game_id	integer(3)	Unique game id, see action available_games	Y
remote_id	varchar	Unique player id	Y
id	integer(3)	Freerounds setup id	Y

args (args[1])	type(max)	description	mandatory Y/N
usr	string	Username for casino login (provided by service provider)	Y
passw	string	Password for casino login (provided by service provider)	Y

**Request:**

[https://<url>/lv8/api/freerounds.js?action=bets&args\[1\]\[usr\]=username&args\[1\]\[passw\]=password&args\[0\]\[game\\_id\]=12&args\[0\]\[remote\\_id\]=123&args\[0\]\[id\]=8](https://<url>/lv8/api/freerounds.js?action=bets&args[1][usr]=username&args[1][passw]=password&args[0][game_id]=12&args[0][remote_id]=123&args[0][id]=8)

**Response:** { "status": 200, "response": "free round removed: 8" }