

Game Client Integration

Introduction

This document describes integration between *RGS (Remote Game Server)* and *Game Client*.

The key words `MUST`, `MUST NOT`, `REQUIRED`, `SHALL`, `SHALL NOT`, `SHOULD`, `SHOULD NOT`, `RECOMMENDED`, `MAY`, and `OPTIONAL` in this document are to be interpreted as described in [RFC 2119](#).

Changelog

Version	Date	Changes
1.0.1	18.06.2020	<ul style="list-style-type: none">Removed nativeld from params
1.0.0	01.06.2020	<ul style="list-style-type: none">First version

Integration

Prerequisites:

- Communication `SHOULD` be done via `HTTP/1.1` and encrypted with `SSL/TLS`.
- All API requests `MUST` be passed with `Content-Type: application/json`.
- All API responses `MUST` be passed with `Content-Type: application/json`.
- Any additional response or request data, parameters, fields or codes not described in documentation `SHOULD NOT` be used.
- Character encoding `utf-8` `SHOULD` be set to `UTF-8`.
- All time and date fields `SHOULD` be in `UTC` timezone.

Authentication

Token authentication (also called bearer authentication) is an [HTTP authentication scheme](#) that involves security tokens called bearer tokens.

The bearer token is a cryptic string, usually generated by the server in response to a login request. The client `MUST` send this token in the Authorization header when making requests to protected resources:

```
Authorization: Bearer <token>
```

Response codes

All responses `MUST` return appropriate response's codes.

Code	Status
2XX	Successful
4XX	Application Error
5XX	Server Error

Errors

All errors **MUST** follow unified structure.

```
{
  "error": {
    "message": "Error description"
  }
}
```

API

Path: /authenticate

Method: POST

Authorization: NO

Authenticate user and start a new session.

Slotify comes with built-in `demo Wallet` and `demo Operator` which will successfully authenticate using any `key` .

Parameters

Name	Type		Example	Description
key	string	REQUIRED	dnsa89me329jdos	Operator's session initialisation key. <code>demo wallet</code> accepts any key.
wallet	string	REQUIRED	myWallet	Wallets's id
operator	string	REQUIRED	myOperator	Operator's id

Response

Name	Type		Example	Description
token	string	REQUIRED	db76b227-0582-45bc-a2cd-fbf38449f28e	Authentication token
balance	float	REQUIRED	100.85	User's current balance
currency	string	REQUIRED	EUR	User's currency, cannot change

Path: /{gameId}/play

Method: POST

Authorization: YES

Creates a new wager.

Parameters

Name	Type		Example	Description
action	string	REQUIRED	gamble	Action

Name	Type		Example	Description
bet	float	REQUIRED	2.58	Cash bet in user's currency. It SHOULD NOT have more than 2 decimal places, otherwise it will be rounded using bankers rounding.
sideBets	object	OPTIONAL	{"double": 5, "triple": 10}	Side bets
roundId	string	OPTIONAL	123e4567-e89b-12d3-a456-426614174000	Round Id for continuation
cheat	string	OPTIONAL	bonus	Cheat for forcing game outcome. Available only in development mode

Response

Name	Type		Example	Description
wager	object	REQUIRED		See Wager
roundId	string	REQUIRED	123e4567-e89b-12d3-a456-426614174000	Round Id used for multi-step rounds
balance	float	OPTIONAL	200.56	User's current balance in user's currency, sent only if changed

Path: `/{gameId}/complete`

Method: POST

Authorization: YES

Completes all transactions and transfers money to user's account.

Parameters

Name	Type		Example	Description
roundId	string	OPTIONAL	123e4567-e89b-12d3-a456-426614174000	Round Id for continuation

Response

Name	Type		Example	Description
balance	float	OPTIONAL	200.56	User's current balance in user's currency, sent only if changed

Path: `/{gameId}/info`

Method: GET

Authorization: YES

Return's configuration and other information

Parameters

None

Response

Name	Type		Example	Description
config	any	OPTIONAL	{"paytable": {...}}	Game config
bets	any	OPTIONAL	{"main": {...}, "gamble": {...}}	Bet config per action. [See Bet Config] (#bet config)
state	any	OPTIONAL	{"collection": 10}	User's game state

Path: `/{gameId}/recover`

Method: GET

Authorization: YES

Returns unfinished rounds

Parameters

None

Response

Name	Type		Example	Description
rounds	Round[]	OPTIONAL		List of rounds See Round

Path: `/{gameId}/cheats`

Method: GET

Authorization: NO

Lists available cheats. Available only in `development` mode.

Parameters

None

Response

Name	Type		Example	Description
cheats	string[]	REQUIRED	["bonus", "any win"]	Cheats

Path: `/{gameId}/history`

Method: GET

Authorization: YES

Returns list of 10 last rounds.

Parameters

None

Response

Name	Type		Example	Description
rounds	Round []	OPTIONAL		List of rounds See Round

Path: `/gameId/replay`

Method: POST

Authorization: NO

Returns round

Parameters

Name	Type		Example	Description
roundId	string	REQUIRED	db76b227-0582-45bc-a2cd-fbf38449f28e	Round Id

Response

Name	Type		Example	Description
round	Round	OPTIONAL		See Round

Types

Round

Name	Type		Example	Description
id	string	REQUIRED	db76b227-0582-45bc-a2cd-fbf38449f28e	Round Id
wagers	Wager []	REQUIRED		List of wagers. See Wager

Wager

Name	Type		Example	Description
data	any	OPTIONAL	<code>[{"spin": 1}, {"spin": 2}]</code>	Game outcome
state	any	OPTIONAL	<code>{"collection": 10}</code>	User's game state persistent between rounds
win	float	REQUIRED	16.45	Cash win in user's currency
sideWins	object	OPTIONAL	<code>{"double": 5, "triple": 10}</code>	Wins from side bets

Name	Type		Example	Description
next	string[]	OPTIONAL	["gamble", "take"]	Next actions for multi-wager rounds. If response contains next array then in next play request action should be set to one of the array's elements.

Bet Config

Name	Type		Example	Description
available	float[]	REQUIRED	[0.01, 0.5, 1, 5, 10]	Available bets in user currency
default	float	REQUIRED	0.5	Default bet in user currency
coin	float	REQUIRED	25	Coin