

# 1 Integrating with IGT RGS

## 1.1 Overview

This guide serves as a comprehensive reference for native systems to integrate with IGT's API endpoints for the iLottery games that are managed by IGT's Remote Game Server (RGS). The RGS solution offers operators an unrivaled desktop and mobile eInstant game library. IGT eInstant content is a new breed of HTML5 entertainment games, driving sales in US states that allow interactive online game solutions for state lotteries (i.e., iLotteries).

## 1.2 Remote Gaming Server (RGS)

One of the most advanced cross-platform delivery systems on the market that powers over 200 branded websites world-wide, RGS provides lotteries with the best in premium game content and plugs in seamlessly to an existing back office. One integration grants access to an extensive library of the world's best-performing games, making them instantly available to your players where and when they want to play: on desktops, mobile phones, and tablet devices.

### 1.2.1 Flexible and Robust Design

The RGS design includes:

- Cloud deployment for optimum performance and scalability as well as local in-state data center deployment to meet different regulatory needs
- Different types of prize and game mechanics
- Reporting services - customized reports from the internal services to provide the best reporting solution/suite as per customer requirements
- Agile product delivery and development with comprehensive out of the box solutions to meet customer requirements
- Native Apps support for faster Apple approvals

### 1.2.2 Exciting Game Features

The IGT games include:

- Fully customizable game console with a cleaner and simple user interface
- Promotional features to offer player bonuses
- Comprehensive progressive jackpot solution
- Support of different game types such as merchandise prize games

The IGT RGS system is a technology solution that serves the IGT portfolio of eInstant games to operators. IGT provides an application service to the native system operator, allowing the administrative overhead to be shared.

---

**Note:** For more information on RGS refer to the [RGS Administration Guide](#).

---

IGT offers a path to an integrated, multi-channel HTML solution for eInstant games. This document focuses on the technical aspects of loading external games on the lottery site, as well

as the needed functional features described in this document to manage the user flow on the external game platform.

With an incremental integration effort, existing RGS partners get unlimited access to all current and future eInstant titles. The IGT RGS system hosts eInstant games in a single HTML framework. The game flow for each game type is similar from the RGS perspective, but the player experience varies.

IGT offers a single set of back-office APIs that support eInstant games. A mutually hosted set of APIs connects the IGT RGS system with the operator's platform.

## 1.3 Hosted APIs v6

### 1.3.1 gameInProgress

This synchronous request from the native system passes parameters that identify the skin and player to determine whether a player has a game in progress in a channel for a certain presentation type.

The native system can use it to query RGS, for example, when changing a player's payment method, which typically is not allowed if a player has a game in progress.

#### 1.3.1.1 API method and path

**GET** /ccs/v1/hostedApi/transaction/getTransaction

#### 1.3.1.2 Header

Both the request and response are in JSON.

Name	Value
Content-Type	application/json
Authorization	Basic {credentials}

#### 1.3.1.3 Request information

#### Request parameters

Parameter Name	Data Type	Parameter Description	Required?
<b>gameType</b>	String	The type of game	Yes
<b>skinCode</b>	String (up to 4 characters)	The unique identifier of a customer configuration in IGT systems	Yes
<b>uniqueId</b>	String (up to 40 characters)	The unique ID for the player (case sensitive)	No

## Request example

```
{
  "gametype": "string",
  "skinCode": "string",
  "uniqueId": "string",
}
```

### 1.3.1.4 Response Information

## Response model

```
gameInProgressList [GameInProgressV6{
  channel                string
  countryCode             string
  currencyCode           string
  currentlyPlaying       boolean
  gameTitle              string
  gameType               string
  languageCode           string
  lotteryGridNumber      integer($int32)
  presentationType       string
  revealId               integer($int32)
  revealVersionId        integer($int32)
  skinCode               string
  softwareId             string
  techCode               string
  txnStartTime           string($date-time)
  wagerAmount            number
}}
status                  string
}
```

## Response parameters

Parameter Name	Data Type	Parameter Description	Required?
<b>channel</b>	String (up to 40 characters)	The channel type of this session: <ul style="list-style-type: none"> <li>• INT: internet/desktop</li> <li>• MOB: mobile device</li> <li>• TAB: tablet device</li> </ul>	No
<b>countryCode</b>	String (ISO 3166)	The country the player provided at registration	Yes
<b>currencyCode</b>	String	The currency the player uses. If the value is not provided, the operator assumes that the game is played in the player's default currency (FPY = Freeplay).	<b>No</b>
<b>currentlyPlaying</b>	Boolean	If true, the player is currently playing an RGS game and the response includes the game title. If false, the player is not playing an RGS game.	No
<b>gameTitle</b>	String	The name of the game being played (not returned if currentlyPlaying = false)	No
<b>languageCode</b>	String (2 or 5 characters)	The two-character code identifying the session language RGS can optionally support a five-character language code. <b>Examples:</b> <ul style="list-style-type: none"> <li>• en (English)</li> <li>• fr-ca (Canadian French)</li> </ul>	
<b>lotteryGridNumber</b>	Integer (\$int32)	The grid ID	Yes
<b>presentationType</b>	String (up to 40 characters)	The presentation type that the game in this session uses: <ul style="list-style-type: none"> <li>• STD: standard</li> <li>• MINI: mini</li> </ul>	No
<b>revealId</b>	Integer (\$int32)	The game's reveal (game mechanic) ID code	Yes
<b>revealVersionId</b>	Integer (\$int32)	The reveal version identifier used to obtain reveal details	Yes
<b>skinCode</b>	String (up to 4 characters)	The unique identifier of a customer configuration in IGT systems	No
<b>softwareId</b>	String	The internal RGS ID for game pay model. More than one <b>softwareCode</b> per game indicates multiple pay models and minBet values.	No
<b>techCode</b>	String	The game's technology HTML FLASH	No
<b>txnStartTime</b>	String (\$date-time)	The current game transaction start time	No
<b>wagerAmount</b>	Number	The total stake of the current ticket	Yes
<b>status</b>	String	A status description of the calling result. It is one of these values: <ul style="list-style-type: none"> <li>• ACCESS_DENIED</li> <li>• OPERATION_DISABLED</li> <li>• INVALID_OPERATOR</li> <li>• UNKNOWN_PLAYER</li> <li>• Success</li> <li>• FAILURE</li> <li>• INVALID SKIN</li> </ul>	No

## Response example

```
{
  "gameInProgressList": [
    {
      "channel": "string",
      "countryCode": "string",
      "currencyCode": "string",
      "currentlyPlaying": true,
      "gameTitle": "string",
      "gameType": "string",
      "languageCode": "string",
      "lotteryGridNumber": 0,
      "presentationType": "string",
      "revealId": 0,
      "revealVersionId": 0,
      "skinCode": "string",
      "softwareId": "string",
      "techCode": "string",
      "txnStartTime": "2023-04-28T17:58:12.096Z",
      "wagerAmount": 0
    }
  ],
  "status": "string"
}
```