

# **Technical Specification**

## **We Are Casino Suite – Operator Integration**

Date: 02/02/2018

Version: 1.6

## Table of Contents

<b>Table of Contents</b> .....	<b>2</b>
<b>1.1 Overview</b> .....	<b>3</b>
<b>1.2 Glossary</b> .....	<b>3</b>
<b>1.3 Starting a game</b> .....	<b>3</b>
1.3.1 Sequence diagram .....	3
<b>1.4 GPI-Publisher communication</b> .....	<b>4</b>
1.4.1 Method Invocation via Generic API.....	4
1.4.2 XML Packet Specifications .....	4
1.4.3 Authentication.....	4
1.4.4 Method parameters .....	5
1.4.5 Return XML Packet .....	5
<b>1.5 Communication API</b> .....	<b>6</b>
<b>1.5.1 Get Player Info</b> .....	<b>6</b>
1.5.2 Request sent from GPI .....	6
1.5.3 Success Response from Publisher .....	7
1.5.4 Failure Response from Publisher .....	7
<b>1.5.5 GetBalance</b> .....	<b>8</b>
1.5.6 Request sent from GPI .....	8
1.5.7 Success Response from GPI.....	8
1.5.8 Failure Response from Publisher .....	9
<b>1.5.9 Bet</b> .....	<b>9</b>
1.5.10 Request sent from GPI .....	9
1.5.11 Success Response from Publisher .....	10
1.5.12 Failure Response from GPI .....	11
<b>1.5.13 Win</b> .....	<b>11</b>
1.5.14 Request sent from GPI .....	11
1.5.15 Success Response from Publisher .....	12
1.5.16 Failure Response from Publisher .....	13
<b>1.5.17 RefundTransaction</b> .....	<b>13</b>
1.5.18 Request sent from GPI .....	13
1.5.19 Success Response from GPI.....	14
1.5.20 Failure Response from GPI .....	15
1.5.21 Error Codes.....	15
1.5.22 Open game URL-Test.....	16
1.5.23 Open game URL-Production.....	16

## 1.1 Overview

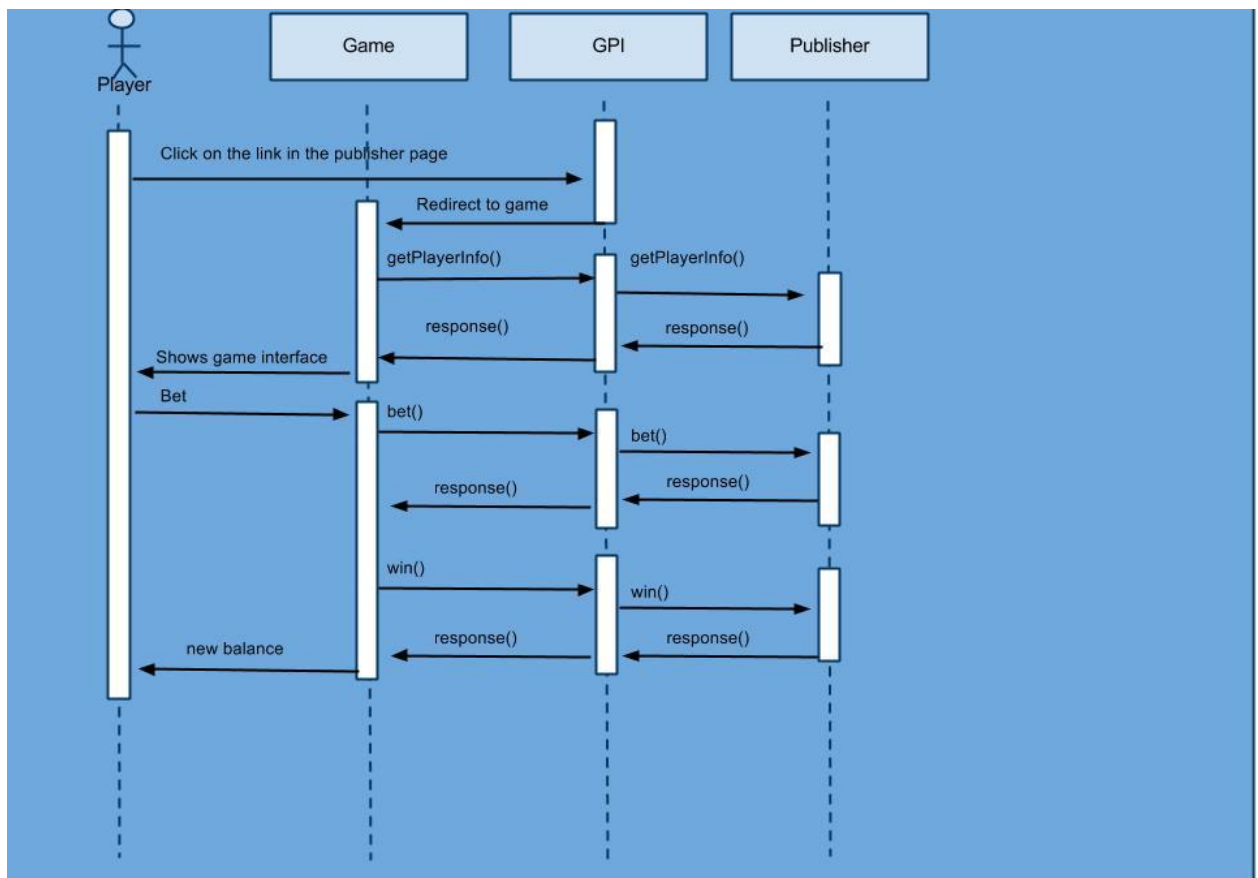
This document explains the proper steps needed to start a publisher integration in our platform.

## 1.2 Glossary

Publisher	The platform where the games will be published
Game	Refers to the game being played by the player.
GPI	The platform responsible to interact between the game and the publisher.

## 1.3 Starting a game

### 1.3.1 Sequence diagram



We provide to publishers a URL, which refers to a GPI service, containing some parameters. When the player start the game, this URL is invoked. Then GPI will redirect to the game URL with the following parameters.

Name	Required	Description
<b>token</b>	true	This <b>token</b> is created by us and is required in the next message
<b>lang</b>	true	This is the language of the player (en, pt, es, etc)
<b>type</b>	false	<b>FREE</b> or <b>CHARGED</b>

## 1.4 GPI-Publisher communication

### 1.4.1 Method Invocation via Generic API

The service uses XML to communicate with the publisher via a HttpWebRequest. The structure of this XML is defined in this document. The XML differs based on the method the service wishes to invoke.

### 1.4.2 XML Packet Specifications

All packets to and from the Publisher are wrapped in a Message Envelope as follows:

```
<message></message>
```

### 1.4.3 Authentication

All requests sent from the GPI should be authenticated by means of a username and password which is attached to the packet as follows:

```
<message>
  <method name="getPlayerInfo">
    <credentials login="login" password="password" />
    ...
  </method>
</message>
```

#### 1.4.4 Method parameters

A method may require parameters to execute. The parameters are specified as follows:

```
<message>
  <method name="getPlayerInfo">
    <credentials login="login" password="password" />
    <params>
      <token value="token" />
    </params>
  </method>
</message>
```

All parameters are wrapped in a `params` node. See the example above. The parameter names are specified in the method definitions contained in this document.

#### 1.4.5 Return XML Packet

Every method call will return a result XML packet that will always contain a `Success` attribute indicating whether the method was successfully executed or not. The packet will contain the result of the method if it has successfully executed. This information is always wrapped in a `returnset` node. For example:

```
<message>
  <result name="getPlayerInfo" success="1">
    <returnset>
      <token value="872de194-8739-47b8-97f8-aa0735f88c52" />
      <loginName value="playerName" />
      <currency value="USD" />
    </returnset>
  </result>
</message>
```

This success return packet contains the result of the `getPlayerInfo` method. The `success` attribute is set to `"1"`. This means it was successfully executed.

If a method has failed to execute, the `success` attribute will contain a `"0"`. The `returnset` will contain a description of the error and an error code as shown:

```

<message>
  <result name="getPlayerInfo" success="0">
    <returnset>
      <error value="Token is invalid or old." />
      <errorCode value="2000" />
    </returnset>
  </result>
</message>

```

The **error** node contains a string with the error message. The **errorCode** node contains a code associated with the error and is of type int.

## 1.5 Communication API

### 1.5.1 Get Player Info

The **getPlayerInfo** method is called when the player login. The method passes a **token**, generated by the Publisher for authentication. The method will return the player's details if the player is authenticated and logged in to the Publisher system.

#### 1.5.2 Request sent from GPI

```

<message>
  <method name="getPlayerInfo">
    <credentials login="login" password="password" />
    <params>
      <token value="token" />
    </params>
  </method>
</message>

```

### 1.5.3 Success Response from Publisher

This method returns the `loginName`, the `currency` and the `balance` in cents.

```
<message>
  <result name=" getPlayerInfo " success="1">
    <returnset>
      <token value="7b3b2cd1-9021-4a38-853b-eec60ed2c180" />
      <loginName value="Tito" />
      <currency value="USD" />
      <balance value="1000000" />
    </returnset>
  </result>
</message>
```

This are the expected params:

token	Token generated by the Publisher.
loginName	Name of the player
currency	The currency of the player
balance	The player balance in cents.

### 1.5.4 Failure Response from Publisher

```
<message>
  <result name="getPlayerInfo" success="0">
    <returnset>
      <error value="Token is invalid or old." />
      <errorCode value="2000" />
    </returnset>
  </result>
</message>
```

### 1.5.5 GetBalance

This packet should be sent to the Publisher to retrieve a player's balance. All amounts are in cents, i.e. 100.00 (USD) will be sent as 10000.

The `getBalance` method should be called whenever needed to update the player's balance in the game interface. Anyway all messages should return the new balance except for the error responses.

### 1.5.6 Request sent from GPI

```
<message>
  <method name="getBalance">
    <credentials login="login" password="password" />
    <params>
      <token value="c9978559-abc8-4c54-a5b6-f23898054c6f" />
    </params>
  </method>
</message>
```

### 1.5.7 Success Response from GPI

```
<message>
  <result name="getBalance" success="1">
    <returnset>
      <token value="c9978559-abc8-4c54-a5b6-f23898054c6f" />
      <balance value="1000000" />
    </returnset>
  </result>
</message>
```

This are the expected params:

token	Token generated by the Publisher.
balance	The player balance in cents.



### 1.5.8 Failure Response from Publisher

```
<message>
  <result name="getBalance" success="0">
    <returnset>
      <error value="Invalid PKT. Method element expected" />
      <errorCode value="-1" />
    </returnset>
  </result>
</message>
```

### **1.5.9 Bet**

This packet is sent to the GPI server for each bet placed in a game. All amounts are in cents, i.e. 100.0 USD) will be sent as 10000.

### 1.5.10 Request sent from GPI

```
<message>
  <method name="bet">
    <credentials login="login" password="password" />
    <params>
      <token value="93998025-f0f5-435f-bc40-36529f6dad18" />
      <transactionId value="1417195937906" />
      <amount value="1000" />
      <gameReference value="flex" />
      <roundId value="1" />
    </params>
  </method>
</message>
```

This are the expected params:

token	Token generated by the Publisher. Received previously from publisher.
roundId	A unique identifier which is unique per player in the Game and can group many transactionId.
transactionId	A unique identifier which references each individual bet in the Game.
amount	Amount to be bet in cents
gameReference	Game name.

#### 1.5.11 Success Response from Publisher

```
<message>
  <result name="bet" success="1">
    <returnset>
      <token value="61311ef9-a80a-4c9a-9a4d-abdfe42c1e19" />
      <balance value="999000" />
      <transactionId value="1417201662161" />
      <alreadyProcessed value="false" />
    </returnset>
  </result>
</message>
```

This are the expected params:

token	Token generated by the Publisher. Will be used in the next request.
transactionId	A unique identifier which references each individual transaction.
alreadyProcessed	If the transaction was processed previously
balance	The player balance in cents.

### 1.5.12 Failure Response from GPI

```
<message>
  <result name="bet" success="0">
    <returnset>
      <error value="Invalid PKT. Method element expected" />
      <errorCode value="-1" />
    </returnset>
  </result>
</message>
```

### **1.5.13 Win**

This packet should be sent to the GPI server for each win which occurs in a game. All amounts are in cents, i.e. 100.00 (USD) will be sent as 10000.

### 1.5.14 Request sent from GPI

```
<message>
  <method name="win">
    <credentials login="login" password="password" />
    <params>
      <token value="63229d8a-b474-4676-9cd3-d6a68f88eb01" />
      <transactionId value="1417457630179" />
      <amount value="1000" />
      <gameReference value="flex" />
      <roundId value="2" />
    </params>
  </method>
</message>
```

This are the expected params:

token	Token received previously from publisher.
roundId	A unique identifier which is unique per player in the Game and can group many transactionId.
transactionId	A unique identifier which references each individual bet in the Game.
amount	Amount to be bet in cents
gameReference	Game name.

#### 1.5.15 Success Response from Publisher

```
<message>
  <result name="win" success="1">
    <returnset>
      <token value="63229d8a-b474-4676-9cd3-d6a68f88eb01" />
      <balance value="999000" />
      <transactionId value="15" />
      <alreadyProcessed value="false" />
    </returnset>
  </result>
</message>
```

This are the expected params:

token	Token generated by the Publisher. Will be used in the next request.
transactionId	A unique identifier which references each individual transaction.
alreadyProcessed	If the transaction was processed previously

balance	The player balance in cents.
---------	------------------------------

#### 1.5.16 Failure Response from Publisher

```
<message>
  <result name="win" success="0">
    <returnset>
      <error value="Error description" />
      <errorCode value="-1" />
    </returnset>
  </result>
</message>
```

#### 1.5.17 RefundTransaction

This packet should be sent to the Publisher server when a refund is required. All amounts are in cents, i.e. 100.00 (USD) will be sent as 10000.

The game sends a **refundTransaction** when the communication fails and there is no knowledge of the status on the Publisher. If the **refundedTransactionId** does not exist in the Publisher then it should respond with an error.

#### 1.5.18 Request sent from GPI

```
<message>
  <method name="refundTransaction">
    <credentials login="login" password="password" />
    <params>
      <token value="a99135e9-3a13-4803-95fd-26be936a4ac9" />
      <transactionId value="1417543910190" />
      <amount value="1000" />
      <gameReference value="flex" />
      <roundId value="1" />
      <refundedTransactionId value="1417543639155" />
    </params>
  </method>
</message>
```

```

    </method>
</message>

```

This are the expected params:

token	Token received previously from publisher.
roundId	A unique identifier which is unique per player in the Game and can group many transactionId.
transactionId	A unique identifier which references each individual transaction in the Game.
amount	Amount to be bet in cents
gameReference	Game name.
refundedTransactionId	It refers to the transaction being refunded.

### 1.5.19 Success Response from GPI

```

<message>
  <result name="refundTransaction" success="1">
    <returnset>
      <token value="a99135e9-3a13-4803-95fd-26be936a4ac9" />
      <balance value="1000000" />
      <transactionId value="5" />
      <alreadyProcessed value="false" />
    </returnset>
  </result>
</message>

```

This are the expected params:

token	Token generated by the Publisher. Will be used in the next request.
-------	---

transactionId	A unique identifier which references each individual transaction.
alreadyProcessed	If the transaction was processed previously
balance	The player balance in cents.

### 1.5.20 Failure Response from GPI

```

<message>
  <result name="refundTransaction" success="0">
    <returnset>
      <error value="Transaction with id=1417543639155 doesn't exists." />
      <errorCode value="5000" />
    </returnset>
  </result>
</message>

```

### 1.5.21 Error Codes

Code	Explanation
0	Internal error
-1	General Exception
100	Unspecified Error
101	The player token is invalid
102	The player token expired
103	The authentication credentials for the API are incorrect
200	Not enough credits
201	Invalid amount
202	Transaction not found

2000	Token is invalid or old
4000	Unknown token
5000	Non existing transaction
9999	Round does not exist

#### 1.5.22 Open game URL-Test

<https://pi-test.njoybingo.com/game.do?token={token}&pn={pn}&lang=en&game={gameCode}&type=FREE>

**{pn}** publisher code generated by WAC.

**{language}** must be replaced with a valid language code. Ex en

**{token}** must be replaced with a valid token generated by you

**{gameCode}** must be replaced with the gameCode of the game you want to open

**{type}** either CHARGED or FREE whether you want to play for free or for real

#### 1.5.23 Open game URL-Production

<https://pi.njoybingo.com/game.do?token={token}&pn={pn}&lang=en&game={gameCode}&type=FREE>

**{pn}** publisher code generated by WAC.

**{language}** must be replaced with a valid language code. Ex en

**{token}** must be replaced with a valid token generated by you

**{gameCode}** must be replaced with the gameCode of the game you want to open

**{type}** either CHARGED or FREE whether you want to play for free or for real