



# **Garena Game Platform**

## **Game Integration Guide**

### **For Garena Partner**

**V 1.05**

**10 October, 2012**

## Change History

NO	Ver	Contents of Change	Who	When
1	0.XX	Created!	LiGeng	2011-12-11
2	1.00	Define the integration standard . Add payment & client related features.	Qunfeng	2012-01-06
3	1.01	Remove the payment gateway part. Add content for token & game client integration.	Qunfeng	2012-03-01
4.	1.02	Reviewed by Zhibo	Zhibo	2012-03-16
5.	1.03	Add IP bonus field in the token.	Qunfeng	2012-04-07
6	1.04	Add token check for multi-usage in Game Server	Qunfeng	2012-04-20
7	1.05	Add the APIs of Game Shop	Qunfeng	2012-10-10

## Table of Contents

<b>1 Overview.....</b>	<b>4</b>
<b>2 User Authentication Integration .....</b>	<b>4</b>
<b>2.1 System Architecture Diagram .....</b>	<b>4</b>
<b>2.2 Integration Flows .....</b>	<b>4</b>
<b>2.3 Token Format .....</b>	<b>5</b>
<b>2.3.1 Encrypted token format.....</b>	<b>5</b>
<b>2.3.2 Plain token format.....</b>	<b>5</b>
<b>2.4 Integration With Game Client/Server .....</b>	<b>6</b>
<b>2.4.1 Launch Game with token .....</b>	<b>6</b>
<b>2.4.2 Post-Process in Game Server Side .....</b>	<b>7</b>
<b>3 Payment Integration.....</b>	<b>8</b>
<b>3.1 Buy Game Currency From Garena Side .....</b>	<b>8</b>

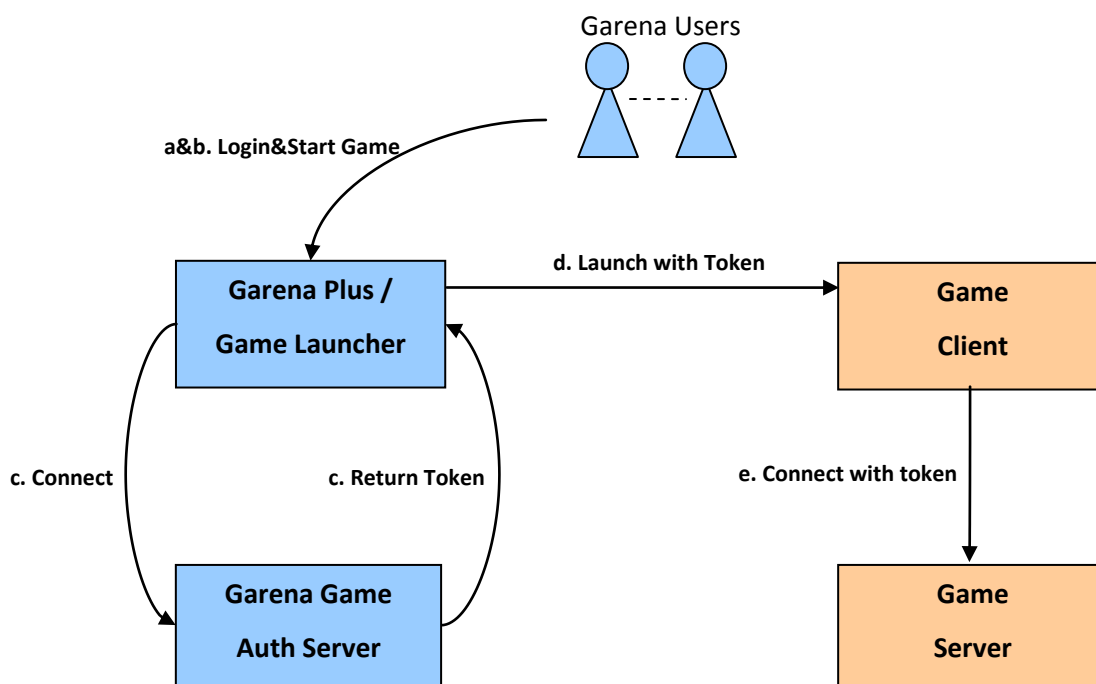
## • Overview

This document describes the integration architecture and specification between Garena and partner games that share user accounts with Garena. User authentication integration and payment integration are the topics in this document.

Once the game is integrated in to the Garena Game Platform, Garena users can launch the game directly from Garena Plus without inputting the game account information again.

## • User Authentication Integration

### .1 System Architecture Diagram



### .2 Integration Flows

The followings describe the steps to launch game from Garena Plus/Game Launcher.

- User login Garena Plus and starts the game on Garena Plus.
- Garena Plus get a encrypted token from Garena Game Authentication Server.
- Garena Plus starts game client with an encrypted token as parameter.
- Game client sends the encrypted token to game server.
- Game server decrypts the encrypted token, and checks user information and timestamp in the plain token.

### .3 Token Format

#### .3.1 Encrypted Token Format

Garena Plus / Garena Game Launcher obtains the encrypted token from Garena Game Authentication Server, and passes this token to the Game Client as a command line parameter. This token is encrypted by RSA:

\* encrypted token = Base64(RSA(plain token))

#### Example of Encrypted Token:

```
sXo+hBxZhXQ1UD0OsJoA5t4q40rrWUzXWcpWbeoJqoogDS4vqY/jwr2l/z6hrLdV6HilG1KudWot85Qe3
rHf/1I52o1ANW673FFXo2k05RniWAAoRgWCcAptepjLyJPW7hQcEuBlqJ/tQQIkAYsjn0sxJG8Q+gNBT/f
n7CMQpWvSulS3Mznv7QkHqtg78mpADqc1T3/AQAFnwgYJ28uzc/6ykWNLx7Sw+lt3EiAejE1sEYzgcip
0Bagq7yl2Eeb4lvvshqIKfKi9l6XZKjn0irqUgy1C9/P6h6WxgA3VLNWh4J0+GlnWDEdQD06kkfFJ23ZC0bo
nGrCdMb9NIUJ8gA==
```

#### .3.2 Plain Token Format

Token has 100 bytes in total (4 + 32 + 4 + 2 + 1 + 57 = 100 bytes) . The details are as follows:

```
=====
|Uid(4bytes)|Name(32bytes)|Time(4bytes)|GCA(2bytes)|GM(1byte)|Bonus(1byte)|Reserved (56bytes)|
=====
```

##### \* Uid :

Garena User ID, 32-bit unsigned integer, 4bytes, big-endian

##### \* Name :

Garena User name, 32bytes, zero-terminated string.

##### \* Time:

The time stamp (UTC) when the token is created, 32-bit unsigned integer, 4bytes, big-endian

##### \* GCA:

Garena Cyber Cafe Alliance Member, 2 bytes. (Range: 0X0000 ~ 0X8000 ), little-endian.

Game specified GCA benefits must be defined by Game Operation Team and Garena GCA Team. Now only 1<sup>st</sup> level common GCA benefits is used in the GCA platform.

The value of GCA should be 0X0001 if a machine in cybercafé has subscribed Garena GCA.

0 ~ 7 bits : common Garena Cyber Cafe GCA benefit.

0 bit : 1 <sup>st</sup> category common GCA benefits. (GCA-GM)	0X0001
1 bit : 2 <sup>nd</sup> category common GCA benefits. (Unused)	0X0002
2 bit : 3 <sup>rd</sup> category common GCA benefits. (Unused)	0X0004
3 ~ 7 bit: Undefined. (Reserved for extension in future)	0X0008 ~ 0X0080

8 ~ 15 bits : Game specified GCA benefits.

8 bit : 1 <sup>st</sup> category game Specific benefits in Cyber Cafe. (Unused)	0X0100
9 bit : 2 <sup>nd</sup> category game Specific benefits in Cyber Cafe. (Unused)	0X0200



10 bit: 3<sup>rd</sup> category game Specific benefits in Cyber Cafe. (Unused) 0X0400  
11 bit ~ 15 bit : Undefined. 0X0800 ~ 0X8000

**\* GM :**

Garena Gold Member for single user, 1 byte.

0: User is not a Gold Member.

1: User is a Normal Gold Mem.ber ( This status is used now. )

2: User is a Titanium Gold Member ( Will be available in future. )

3: User is a Diamond Gold Member ( Will be available in future. )

>3: Reserved for Future extension. Should include all benefits of the 3 statuses.

**\* Bonus :**

Special benefits/bouns if user login from a special IP, 1 byte.

0: User does not have any bonus.

1: User has IP bonus. ( User can have some benefits if the user login from a specified IP.)

>1: Reserved for future extension.

**\* Reserved :**

Reserved data, 56 bytes ( Only static content is supported. For example: Garena )

## .4 Integration With Game Client/Server

### .4.1 Launch Game with Token

Garena Plus/Game Launcher start the game after getting an encrypted token from Garena Game Authentication Server. This token will be passed to the Game client by parameters.

**SYNTAX:**

➤ <Game Client Name> [parameter\_string1]**encrypted\_token**[parameter\_string2]

\* Game Client Name: EXE file name to launch a game,

\* parameter\_string1: The parameter string before the **encrypted token**, Optional.

\* parameter\_string2: The parameter string after the **encrypted token**, Optional.

**NOTE:**

- encrypted\_token: please refer to 2.3.1 for the format/content.
- Only the encrypted\_token is generated dynamically by Garena Game Authentication server.
- parameter\_string1 & parameter\_string 2 are optional.
- The content of parameter\_string1 & parameter\_string2 is decided by Garena Partner.  
Once decided, the content of these two parameters **can not be changed**. The game plugin needs to be updated if we want to change the content of these two parameters.

- e. Garena Partner is responsible for parsing parameter\_string1 & parameter\_string2.
- f. A specific a name for each parameter is recommended to avoid error when parsing parameters. Some examples are listed below (a -d )

#### EXAMPLES:

- a. XXXClient.exe param1\_n:param1\_v token:encrypted\_token ...paramX\_n:paramX\_v
  - \* Game Client Name: "XXXClient"
  - \* parameter\_string1: "param1\_n:param1\_v token:"
  - \* parameter\_string2: " ...paramX\_n:paramX\_v"
- b. LoL.exe partner:garena token:encrypted\_token gameid:8001
  - \* Game Client Name: "LoL.exe"
  - \* parameter\_string1: "token: "
  - \* parameter\_string2: " gameid:8001"
- c. XXX.exe encrypted\_token garena 8009
  - \* Game Client Name: "XXX.exe"
  - \* parameter\_string1: "", null string
  - \* parameter\_string2: " garena 8009"
- d. YYY.exe encrypted\_token
  - \* Game Client Name: "YYY.exe"
  - \* parameter\_string1: "", null string
  - \* parameter\_string2: "", null string

#### .4.2 Post-Process in Game Server Side

For security purpose, the encrypted token should be decrypted in Game Server side.

Game Server need to do the following steps to get the user's information from the token and verify the token's validity.

- a. The token can only be used **once** in 5 minutes. If this token has been used in 5 minutes, then the token should be discarded.
- b. Decrypt the encrypted token by using a **public key** provided by **Garena**.
- c. Get the timestamp from the token and check the expiry of the token. A token is only valid in 30 minutes.

- **Payment Integration**

## **.1 Buy Game Currency Through Garena**

For a specific game, a Game Currency Shop will be created at the Garena Web Site. Users can buy the game currency by using Garena Shells in the Garena Shop. For example:

- \* <http://buyrp.garena.com/>
- \* <http://rp.lol.garena.tw/>
- \* <http://shop.pw2.garena.com>

Garena Partner will need to provide the interface, using which Garena can add the currency in game for Garena Users.

## **.2 API for GameShop**

### **.2.1 API - Check User Information**

This API is used to check the “Character” or “Name” used by our gamer in the game.

Game shop need to check whether this user have created the necessary account or characters in the game before they buy the game currency.

Recommended parameters for this API.

- garena\_uid: UID of user’s Garena Account, this parameter is optional if only “Garena Name” is used by game.
- garena\_name: Name of user’s Garena Account, this parameter is optional if only “Garena UID” is used by game.
- Other parameters: Game Factory can decide other parameters which are necessary.

Recommended return result for this API:

- The “game account name” / “character name” / “charcter list” of the Garena User.
- The “Balance Of In-Game Currency” if possible. ( Add a new API for this feature is OK)



## .2.2 API - Add Game Currency

This API is used to add the “game currency” for users when they buy the “In-Game currency” in Garena Game Shop.

Recommended parameters for this API.

- `game_account/character_name`: The in-game account/character retrieved in 3.2.1
- `currency_amount`: amount of the game currency to be added.
- `server_id`: Server identifier if necessary.
- `garena_uid`: UID of user’s Garena Account.
- `garena_name`: Name of user’s Garena Account, this parameter is optional if only garena UID is used by game.
- Other parameters: Game Factory can decide other parameters which are necessary.

Recommended return result for this API:

- The “game account name” / “character name” / “charcter list” of the Garena User.

## .2.3 Examples of API

//check the exist of role in game

**checkUsername( log\_uid, log\_uname, log\_reason)**

- `log_uid`: the login id for garena
- `log_uname`: the login name for garena
- `log_reason`: the login reason

//convert the shell to game coin

**topup(username, zoneid, cash, log\_uid, log\_uname, log\_reason)**

- `username`: the name of user
- `zoneid`: the game server id
- `cash`: the amount of game coin
- `log_uid`: the login id for garena
- `log_uname`: the login name for garena
- `log_reason`: the login reason