



CREATIVE GAMING

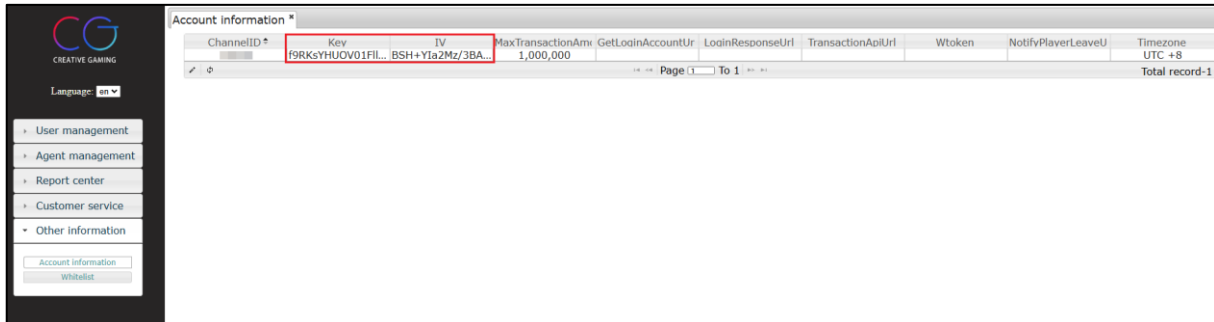
# Technical Document

Chapter 2 Starting The Game

# Chapter 2 、 Starting The Game

## 一 、 Encryption and decryption

1. Take the key and vector parameters (iv & key) in the background



Note: The length of the key is 44, and the length of the iv is 24.  
Please confirm the Length of the string before use.

2. Use the appropriate sample code for encryption and decryption according to your platform environment.

## 二 、 Create player account

1. Build a Json string, for example: {"accountId":"UserID000001","currency":"CNY"}

Description:

- "UserID000001" is the name of the player account to be created.
- " CNY" is the currency type code of the channel number. If you fill in the wrong currency, CG will return an error code.

2. Encrypt {"accountId":"UserID000001","currency":"CNY"} to get a similar string:

[KO8FNO/omYMHVlcsg6G2U6MPhKuVZnB4SOXWj9OQGw6HMDusCB3G977ligX3LFH](#)

3. Using the POST request

- header fills in the Content-Type content is empty
- URL is: [http://testenv-dev.cg11systems.com:8088/td\\_create\\_account](http://testenv-dev.cg11systems.com:8088/td_create_account)

- Body uses the raw text content is

version=1.0&channelId=92492&data=KO8FNO/omYMHVlcsg6G2U6MPhKuVZnB4SOX  
Wj9OQGw6HMDusCB3G977ligX3LFHS

- Request method example:

POST ▼ http://testenv-dev.cg11systems.com:8088/td\_create\_account

Params Authorization Headers (10) **Body** ● Scripts Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **Text** ▼

```
1 version=1.0&channelId=63395&data=Csje9jCeHwqPX9Ebe0MAkUdxu596vYHHm1XxHrFpPPZB19uCixjTCrkjzim95uNk
```

POST ▼ http://testenv-dev.cg11systems.com:8088/td\_create\_account

Params Authorization **Headers (10)** **Body** ● Scripts Tests Settings

Headers ◀ 9 hidden

	Key	Value
<input checked="" type="checkbox"/>	Content-Type	
	Key	Value

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 BZ0AS0CcukwN89ppV10nDL+mPezxS/MUt+GzK5pmMao=
```

Description:

- version=1.0 is a fixed string must be filled in
- channelId Please adjust to the correct channel number
- Data is a Json string, The encryption result of

```
{"accountId":"UserID000001","currency":"CNY"}
```

```
KO8FNO/omYMHVlcsg6G2U6MPhKuVZnB4SOXWj9OQGw6HMDusCB3G977ligX3LFHS
```

※Please adjust the content according to the actual situation.

4. Receive response string of CG:

```
18ZUWV7NdY9zKcNqxeh2feDsTI/McwQoY7t3jXx1Q48=
```

Available after decryption: {"errorCode":"0"}

5. If the errorCode is not 0, · please refer to the error code of

“CG technical document\_Chapter07\_Refer table” for comparison.

**Note: Before requesting to create a player account, you need to add it to the whitelist before the CG server will release it and respond to your end.**

### ≡ 、 Slot game login

1. Design a login verification string by yourself, example: cg\_login\_token

2. If you are using CG for the first time, it is recommended to use a fixed string.

After the completion of the concatenation, change to the dynamic string as required.

3. Encrypt the login authentication string with iv & key, such as cg\_login\_token,

After encryption, you will get string like:

```
dVJsiXCFTFYzMp6UEfGpUw==
```

4. Encrypt the encrypted string into UrlEncode, Original encrypted string:

```
dVJsiXCFTFYzMp6UEfGpUw==
```

Get it after using UrlEncode: [dVJsiXCFTFYzMp6UEfGpUw%3D%3D](#)

5. Combine the verification string with the startup game URL to get:

<https://static.exonova-logic.com/cgtest/LuckyPig/?version=1.0&language=cn&channelId=66316&data=dVJsiXCFTFYzMp6UEfGpUw%3D%3D>

Description:

- i. LuckyPig is the name of the game to start, please refer to the excel file for a list of available.
- ii. version=1.0 is fixed must be filled in.
- iii. data is the string for login verification.

**Request available parameter list:**

Name	Description	Type	Allow empty
version	Version number	String	N
language	default: cn cn: Simplified Chinese tw: Traditional Chinese en: English th: Thai kr: Korean vn: Vietnamese pt: Portuguese	String	Y
channelId	Channel number	String	N
homeUrl	Home link, please handle with UrlEncode	String	Y
data	Encrypt content, log in to the token string for the player	String	N
noSleep	Anti-sleep function activated on mobile device true: open (default) ;false: off PC device defaults to false	String	Y

mask	iOS mobile device (including iPhone, iPad) triggers full-screen sliding tips, true: open (default) ; false: off	String	Y
showHome	Show home button true: open ; false: off (default)	String	Y
coinSymbol	When the url has this parameter, if the value is not a null string, CG will replace the coin symbol with this string.	String	Y
debug	Display log in the console true: open ; false: off (default)	String	Y
serverDebug	Display the server log in the console true: open ; false: off (default)	String	Y
mute	Turn off music and sound true: open ; false: off (default)	String	Y
fullScreen	full-screen display true: open (default) ; false: off	String	Y
homeCallback	When the home button is clicked, the javascript assigned by homeCallback is called.	String	Y
dialogStyle	Dialog display mode 0: display (default) ; 1: not display	Number	Y

6. If the game is started correctly, CG will send a verification request to

[GetLoginAccountURL] registered in the Chapter01.

7. Request format is HTTP POST

- Header use Content-Type = application/x-www-form-urlencoded

The parameters are as follows:

- version=1.0, This is a fixed value.
- channelId=66316 · This is the channelId brought in the URL of the previous startup game.
- data=n1YuCJnzzhLTlJmKR2UOZj26aRf15Ks33jMMbIxhhNs=  
This is the Json string {"token":"cg\_login\_token"} the result of the encryption.
- CG will decrypt the data brought in the startup game URL to get the token string: cg\_login\_token
- Then CG will make Json's packaging become {"token":"cg\_login\_token"}
- After encrypting the Json string, you will get it:

n1YuCJnzZhLTLJmKR2UOZj26aRf15Ks33jMMbIxhhNs=

8. The logical implementation of decryption and player identity verification on the [GetLoginAccountURL] that you provide to CG.

If you use CG for the first time, it is recommended to respond directly to the correct content, without making decryption and verification logic,

After the serial connection is completed, it is adjusted to the flow with inspection logic according to the platform requirements.

9. Encrypt the Json string before responding, such as:

FYlIRdUBZSh8COFHqXPi0BL9+A5dGQ9HpiBB0pOGuA4Mo6tXqp8iuLUXFctj8K3P2psWf  
mr/g2m7YwjIWklwv79xAX+ActirDUu7WYKaHeCk0Vh0MYPmTJ2AxxyrK25t

10. If the logged in player status is verified, you need to respond to the content as:

```
{"channelId":"66316","accountId":"UserID000001","nickName":"nickName",  
"errorCode":0}
```

The above are only necessary parameters for the content to be returned. For the other parameters, please refer to " CG technical document\_Chapter09\_ Set the Betting Limits and Prevent game URL tampering "

**parameter content:**

Name	Description	Type
channelId	Channel number	String
accountId	Player unique identifier (account), consisting of English letters and numbers (case insensitive)	String (127)
nickName	Nickname (will be displayed in the upper left corner of the game client to identify the player)	String (127)
errorCode	Error code number	Number

Description:

1. channelId is the value brought by the CG request
2. accountId is the correct account that must be filled in after the your side check
3. nickName is designed by your side' s member system, fill in according to the actual situation
4. errorCode is the identifier of the CG execution result. If the value is not 0, refer to the error code at the CG technical document\_Chapter07\_Refer table.
5. **If you use transfer wallet, you can start the game correctly now.**



6. If you use seamless wallet, the game still not work here, you need complete "CG technical document\_Chapter03\_Seamless Wallet"
7. If an error occurs during startup, please take a screenshot of the startup error screen and provide the URL to start the game to the CG technical team. We will assist you as soon as possible.