

# Game Server API Document

Name	Date	version	remark
Qingping	2020.3.17	1.0.0	created
Qingping	2020.4.22	1.0.1	remove analytic data api
Qingping	2020.4.23	1.0.2	use new api url
Qingping	2020.7.8	1.0.3	add new api
Qingping	2020.7.10	1.0.4	add new api
Qingping	2020.7.15	1.0.5	document format
Qingping	2020.8.3	1.0.6	add error code
Qingping	2020.8.26	1.0.7	add transfer api
HanPeng	2020.9.23	1.0.8	add exchange order api
HanPeng	2020.11.26	1.0.10	add sync user game zone data api
HanPeng	2020.12.9	1.0.10	modify sync user game zone data api add sync order game zone data api

## 1、Basic

test : <https://t-api.amusgame.net>

production: <https://api.amusgame.net>

Every API must be called in HTTP post method, with Content-Type:  
application/x-www-form-urlencoded

## 2、Signature

Every api request must contains common parameters below:

name	type	description
appId	string	your game app's id,generated by GameCenter

timestamp	long	request time, milliseconds since 1970-01-01
nonce	int	a random number, be different from nonce in the last request
signature	string	all parameters signature

#### **signature generate method:**

first, sorted all parameters by parameter name in asc order.

second, concat the sorted parameters , paramStr = "key1=value1&key2=value2"

last, signature = sha256(paramStr + <server secret>)

sha256: a SHA-256 encrypted hash generator method, it is easy to implement with C#,Java,PHP etc.

<server secret>: generated by GameCenter

## **3、Auth APIs**

### **Check user access token**

#### **request:**

*POST /gameserver/auth.check\_status*

*userId=293838182132212&userToken=213k3j3321k2k32j33k4k4*

the *userToken* is accessToken in GameCenter SDK

#### **response:**

```
{
  "code": 0,
  "msg": "success",
}
```

*code* is 0, check success

*code* isn't 0, check failed, may be illegal userToken or userId

## 4、SNS & User APIs

### Get all friends

**request:**

```
POST <domain>/gameserver/user.friends
Content-Type:application/x-www-form-urlencoded
userToken=213k3j3321k2k32j33k4k4
```

**response:**

```
{
  "code":0,
  "msg":"success",
  "data":[
    {
      "photoUrl":"https://totok-app-test.oss-me-east-1.ali
yuncs.co/48c8f8837d2414d8736b479c45a91ebe.jpeg?Expires=1594209
577&OSSAccessKeyId=LTAIJWkUjWgW8nc&Signature=%2BLc%2FFVogjQQi
RoRps7QPgJzx30A%3D",
      "ttkUid":86179877528939918,
      "nickname":"师元君"
    }
  ]
}
```

*ttkUid* is difference from *userId*, *ttkUid* is used to GameCenterSDK sharing API

## Get playing friends

### request:

```
POST <domain>/gameserver/user.playing_friends
Content-Type:application/x-www-form-urlencoded
```

```
userToken=213k3j3321k2k32j33k4k4
```

### response:

```
{
  "code":0,
  "msg":"success",
  "data":[
    {
      "photoUrl":"https://totok-app-test.oss-me-east-1.ali
yuncs.co/48c8f8837d2414d8736b479c45a91ebe.jpeg?Expires=1594209
577&OSSAccessKeyId=LTAIJWKkUjWgW8nc&Signature=%2BLc%2FFVogjQQi
RoRps7QPgJzx30A%3D",
      "userId":86179877528939918,
      "thirdUid":86179877528939918, //totok uid
      "thirdType":1, // thirdType=1 means totok user
      "nickname":"师元君"
    }
  ]
}
```

## Sync ranking data

synchronize the ranking data of players playing the game

**request:**

*POST* <domain>/gameserver/sync.rank\_data

*rank\_type*=<rankType>&*rank\_data*=<rankData>

<rankType>: an identifier for game ranking type.

<rankType>	comment
0	game level
1	game score
2	game checkpoint

<rankData>: a json string for a pair list,

[[<userId>,<value>],[<userId>,<value>],[<userId>,<value>]]

<userId> is GameCenter platform user's id, it's callback by GameCenter SDK auth method

<value> is the value for the ranking type, a number

for example:

[[1232938202023,23],[139328288232212,12322],[12384829399444,1]]

**response:**

```
{  
  "code": 0,  
  "msg": "success"  
}
```

**limited:**

less than 1000 users rank data in one api request .

one user rank data only one submit every day .

user rank data need not submit if it is no changed .

## Sync user game zone data

synchronize the game zone data of players playing the game

### request:

*POST* <domain>/gameserver/sync.user\_game\_zone\_data

*user\_game\_zone\_data*=<userGameZoneData>&*user\_game\_zone\_type*=<userGameZoneType>

<userGameZoneData>: a json array string,

```
[[<userId>,<gameZone>,<roleCreateTime|roleActiveTime>],[<userId>,<gameZone>,<roleCreateTime|roleActiveTime>],[<userId>,<gameZone>,<roleCreateTime|roleActiveTime>]]
```

<userGameZoneData>:

1: create

2: active

<userId> is GameCenter platform user's id, it's callback by GameCenter SDK auth method

<gameZone> is the value for the game zone, a number[0,32767]

<roleCreateTime> role create timestamp, milliseconds since 1970-01-01

<roleActiveTime> role active timestamp, milliseconds since 1970-01-01

for example:

*user\_game\_zone\_data*=[[1,1,1],[2,2,2],[3,3,3]]&*user\_game\_zone\_type*=1

### response:

```
{
  "code": 0,
  "msg": "success"
}
```

**limited:**

no more than 1000 eles in *userGameZoneData* in one api request .

only create role event in one game zone need submit immediately.

one create role event need submit only once.

## Sync order game zone data

synchronize the game zone data of players order

**request:**

*POST <domain>/gameserver/sync.order\_game\_zone\_data*

*order\_game\_zone\_data=<orderGameZoneData>*

*<orderGameZoneData>*: a json array string,

*[[<orderId>,<gameZone>],[<orderId>,<gameZone>],[<orderId>,<gameZone>]]*

*<orderId>* is callback by the SDK payment method

*<gameZone>* is the value for the game zone, a number[0,32767]

for example:

*order\_game\_zone\_data=[[1,1],[2,2],[3,3]]*

**response:**

```
{
  "code": 0,
  "msg": "success"
}
```

**limited:**

no more than 1000 eles in *orderGameZoneData* in one api request .

one create order event need submit only once immediately.

## ***5, Order & Pay & Transfer APIs***

### **Check order status**

**request:**

**POST** <domain>/gameserver/order.check\_status

orders=<orderId1>,<orderId2>

the <orderId> is callback by the SDK payment method

**response:**

```
{
  "code":0,
  "msg":"success",
  "data":{
    "144612098616033280":{
      "checked":0,
      "gameId":126847495400849409,
      "orderId":144612098616033280,
      "payTime":0,
      "amount":100, //unit cents
      "currency":"AED",
      "sku":"001",
      "status":0 // not pay
    },
    "141641699233734684":{
      "checked":1,
      "gameId":126847495400849409,
      "orderId":141641699233734684,
```



```

        "payTime":1582019617910,
        "amount":100, //unit cents
        "sku":"001",
        "currency":"AED",
        "status":1
    }
}
}

```

status=1, order has been payed

status=0, order isn't payed yet

status=3, order has been reset,may be a illegal order

checked=0, order status is not the final status

checked=1, order status is the final status

if status=1 , items of the order should be sent to user's account

if checked=1 and status=0 or status=3 , the order should be forced to cancel

## ***add exchange order***

*when player exchanging the code get from khalaspay, record the purchase order*

### **request:**

*POST <domain>/gameserver/order.new\_exchange*

*userId=XXXX&channel=5&env=XXXX&sku=XXXX&orderRef=XXXX&amount=xxxx&currency=U*

*SD*

userId:is GameCenter platform user's id, it's callback by GameCenter SDK auth method

channel:"5" fixed

env:payment environment.0,sandbox;1,production

sku:product id

orderRef:transaction id

amount: price, accurate to 2 decimal places, like:"100.23"

currency: "USD" fixed

**response:**

```
{
  "code":0,
  "msg":"success",
  "data":"566422073705434171" //orderId
}
```

## ***Create&submit transfer order***

*transfer from merchant account to user's wallet*

**request:**

*POST <domain>/gameserver/transfer.new*

*userToken=213k3j3321k2k32j33k4k4&amount=xxxx&currency=AED*

amount: withdraw amount, accurate to 2 decimal places, like:"100.23"

currency: "AED" fixed

**response:**

```
{
  "code":0,
  "msg":"success",
  "data":"566422073705434171" //transferId
}
```

## Check transfer status

**request:**

**POST** <domain>/gameserver/transfer.check\_status

transfers=<transferId1>,<transferId2>

the <transferId> is callback by the "Create&submit transfer order"

**response:**

```
{
  "code":0,
  "msg":"success",
  "data":{
    "141641699233734684":{
      "user_id":645633159128785116,
      "gameId":126847495400849409,
      "transferId":141641699233734684,
      "amount":100, //unit cents
      "currency":"AED",
      "status":1
    }
  }
}
```

status=0, transfer created

status=1, transfer submitted to gateway

status=2, transfer success

status=3, transfer failed

## 6、Error codes

code	msg
------	-----

0	success
900	internal error
920	unauthorized request, need login.
921	<pre>{"reason":"","expired":milliseconds}</pre> <p>this code means user was blocked</p>
922	unauthorized request, invalid access token
923	no permission
924	invalid access token, another client was authorized.
998	duplicate operation
999	invalid parameter
370004	transfer failed, rejected by pay gateway