



# JDB Platform API Specification

Version: 2.6.30  
Date: 2021-05-18

## Revision History

Date	Version	Revised Items	Revised by
2016-09-08	2.1	<ul style="list-style-type: none"> <li>i. Get Token: Add parameter “windowMode”.</li> <li>ii. Create user: Change “credit_allocated” into optional field.</li> <li>iii. Search for running text message: Remove the return parameter “target”.</li> <li>iv. Withdraw/Deposit: Change the input data “allCashOutFlag” into 0, 1.</li> <li>v. Search for details of transactions in slot games and in sports betting: Change the response time of the latest information grab into 3 minutes from requested time.</li> <li>vi. The parameter “parent”(parent ID) is added to the following functions: search for user data, kick out user, cash transfer history, check cash transfer, force settlement, change mobile user’s login password, Jackpot Contributions on/off flag (action: 15, 17, 18, 28, 33, 36, 43)</li> <li>vii. The parameter name “agentId” is changed into “parent” when search for details of game history transactions in slot games (action: 29)</li> <li>viii. Add new arcade machine code: 45,51</li> </ul>	Brenda Hou
2016-09-29	2.2	<ul style="list-style-type: none"> <li>i. Add: Search for details of game history transactions in slot games By House(action 44)</li> </ul>	Brenda Hou
2016-10-26	2.3	<ul style="list-style-type: none"> <li>i. Add: Daily Report (action 42)</li> </ul>	Brenda Hou
2016-10-28	2.4	<ul style="list-style-type: none"> <li>i. Add: Search for Jackpot information(action 45)</li> <li>ii. Add PHP program samples</li> </ul>	Brenda Hou
2016-11-18	2.4.1	<ul style="list-style-type: none"> <li>i. Add description for the parameter length</li> </ul>	Brenda Hou
2016-12-21	2.4.2	<ul style="list-style-type: none"> <li>i. Add new language: Vietnamese</li> </ul>	Brenda Hou
2017-01-04	2.4.3	<ul style="list-style-type: none"> <li>i. Adjust Daily Report: Add count column.</li> <li>ii. Correct Daily Report: The parameter name “gtype” is revised to “gType”.</li> </ul>	Brenda Hou
2017-02-08	2.4.4	<ul style="list-style-type: none"> <li>i. Add demo account trial function</li> <li>ii. Adjust Daily Report: The “bet” will always display as negative values.</li> <li>iii. Add new arcade machine code–Slot Game:</li> </ul>	Brenda Hou

		LUCKY MINER, CANDY LAND, SUPER DUMPLING	
2017-03-06	2.5	i. Add: Search for details of game history transactions in fishing games (action 46) ii. Add new currency: Indian rupee (INR) iii. Daily Report: Add fishing game machine	Brenda Hou
2017-03-21	2.5.1	i. When in Single Window Mode, client side can choose whether to hide the Home button by setting.	Brenda Hou
2017-04-13	2.5.2	i. Adjust Get Token and Get Demo Token as well as the field description of "windowMode". ii. Arcade machine code: Add "Slot Game for HTML 5"	Brenda Hou
2017-04-20	2.5.3	i. Remove: Arcade machine codes in Appendix	Brenda Hou
2017-06-05	2.5.4	i. Withdraw/Deposit: Add function notes ii. Search for details of game history transactions in slot games(action 29, action 44): Add the return parameter "gambleBet" iii. Search for details of game history transactions in slot games(action 29, action 44):, Search for details of game history transactions in fishing games(action 46): Remove the return parameter "gameName"	Brenda Hou
2017-07-01	2.5.5	i. action 11 Get token: Add the return parameter "path"	Roger Wu
2017-07-04	2.5.6	i. action 11 Get token: Delete the return parameter "x" ii. action 47 Get Demo token: Add the return parameter "path"; delete the return parameter "x" iii. Adjust the description of login system	Brenda Hou
2017-07-04	2.6.0	i. Remove the search for details game history transactions in fishing games (action 46). ii. Adjust action 29: Provides all details game history transactions in games and add gType to sort by game type.	Brenda Hou
2017-08-10	2.6.1	i. Adjust action 29: Add information of game history transaction in Bar games ii. Update action 42: Add Bar games	Brenda Hou

2017-09-04	2.6.2	i. Update action 42: Add “validBet” data	Roger Wu
2017-11-07	2.6.3	i. Update action 11, action 47: Add check available of gType + mType, Verification failed to return error code: 8000 ii. Adjust action 12, action 13: Name checking for no special characters allowed <>#'"%-=*/ & iii. Update action 28: add in balance after deposit / withdraw (afterBalance) iv. Adjust action 40, action 41: change the content length limit to 100 words String(100), instead of only grab first 100 words v. Add: Query Game List (action 49) vi. Add Error Code: 1. 7405: You have been logged out! 2. 8010: The length of parameter is too long.	Amuro Huang
2017-11-08	2.6.4	i. Add: Query In-Game User (action 52) ii. Add Error Code: 1. 7505: User is not playing	Amuro Huang
2017-11-14	2.6.5	i. Update action 11, action 47: Add parameter “isAPP”	Roger Wu
2017-11-17	2.6.6	i. Update action 11, action 47: Add parameter “lobbyURL” ii. Correct the wrong example	Amuro Huang
2017-11-24	2.6.6	i. Update action 52, add a parameter “balance”	Roger Wu
2017-11-27	2.6.7	i. Add action 54: Query Game Result ii. Update action 12,13: name String(50) iii. Update action 15: username String(50)	Roger Wu
2017-12-26	2.6.8	i. Adjust action 19, 28: serialNo String(50) ii. Add action 55: Check Cash Transfer (Same as action 28, and it will return error code 9015 when search result is no data) iii. Remove: 1. Search for Running News (action 40) 2. Update Running News (action 41)	Amuro Huang
2018-03-28	2.6.9	i. Add program samples: C#	Brenda Hou
2018-04-03	2.6.10	i. Adjust action 29: fishing game (gType = 7) ii. add field roomType (2: CaiShen Area) iii. Add action 58, Kick out downline users	Amuro Huang

		iv. Add Error Code: 1. 9017: Work in process, please try again later.	
2018-07-11	2.6.11	i. Add FTP Function ii. Update action 11, action 47: Add parameter “moreGame”, “mute” iii. Update action29: Add Lottery Game Type(gType = 12) iv. Add Error Code: 6901	Brenda Hou
2018-08-23	2.6.12	i. Update action 54: Add “showUid” ii. Update action11: Add parameter “cardGameGroup” iii. Update action29: Add Card Game Type(gType = 18) iv. Update action42: Add api return parameter “tax” If card game is enable to play then call api by different game type will get the api response with tax parameter	Brenda Hou Kasim Guo
2018-08-30	2.6.13	i. Adjust Action 44: Provides all details game history transactions in games ii. Add Action 61: Cash Transfer History (Including Rewards) iii. Add Action 62: Check Event Ranking iv. Delete Action 16,27	Brenda Hou
2018-11-06	2.6.14	i. Update action 29,44 1. “seqNo”, “roundSeqNo” return format changed from Integer to Long ii. Max length of player id changed from String(19) to String(25)	Brenda Hou Amuro Huang
2019-03-20	2.6.15	i. Enhance document description ii. Remove Action 13: Update User iii. Remove Action 18: Cash Transfer History Please use Action 61 iv. Remove Action 28: Check Cash Transfer Please use Action 55 v. Remove Action 33: Force Settlement vi. Remove Action 36: Change Mobile User’s	Brenda Hou

		Password vii. Fix the error code in English: 8007, 7601 viii. Increase the currency code ix. Sample program adds Go Lang version.	
2019-05-03	2.6.16	i. i. Adjust Action 29,44: add “validBet” data to betting records in Card Games	Kasim Guo
2019-05-08	2.6.17	i. i. Adjust Action 11: cardGameGroup parameter format and description	Kasim Guo
2019-09-18	2.6.18	i. Sample program adds C# decrypt sample ii. Fix php sample iii. Fix action 14 function description iv. Action 11, 47 add field isShowDollarSign	Amuro Huang
2019-10-02	2.6.18.1	i. Adjust Action 49: Modify description	Amuro Huang
2019-11-06	2.6.19	i. Update action 29,44 “gType” return format changed from String(2) to Integer	Amuro Huang
2020-05-06	2.6.20	i. Update action 19 : description and sample code ii. Sample program update PHP decrypt sample.	Amuro Huang Brenda Hou
2020-09-01	2.6.21	i. Action 29, 44 Betting records in fishing games- Modify roomType description. ii. Action 44 Betting records in slot games - lastModifyTime format change to String(19) iii. Action 15,29,44: Max length of currency changed from String(2) to String(3) iv. Currency code add remark: ratio	Amuro Huang Kasim Guo Brenda Hou
2020-12-10	2.6.22	i. Add Action 64 Search for Old Details of Transactions in Games ii. Adjust Action 29 function description iii. Add JavaScript program sample	Kasim Guo Joshua Wang Angus Wu
2021-01-07	2.6.23	i. Adjust Action 64 data provide start date	Kasim Guo Joshua Wang
2021-02-03	2.6.24	i. Action 65: Query Online Users with balance > 0	Chester Young
2021-02-23	2.6.25	i. Modify roomType description in Action 29, 44, 64	Angus Wu Shadow Sun
2021-03-24	2.6.26	i. Modify x description	Kai
2021-04-09	2.6.27	i. Add Action 19 function description	Brenda Hou
2021-04-27	2.6.28	i. Action 11、 47 language add Thai and Vietnamese	William Ezno

2021-05-14	2.6.29	i. Add BRL currency	Young
2021-05-18	2.6.30	i. Add CLP currency	Brenda Hou

## Table of Contents

1.	Precautions and process description .....	3
1.1.	Precautions.....	3
1.2.	Required function description .....	4
2.	Functions.....	5
2.1.	Action 11: Obtain Token .....	5
2.2.	Action 12: Create Player .....	8
2.3.	Action 14: Player Status Management.....	10
2.4.	Action 15: Search player information .....	12
2.5.	Action 17: Kick Out Player .....	14
2.6.	Action 19: Withdraw / Deposit .....	16
2.7.	Action 29: Search for Details of Transactions in Games .....	19
2.8.	Action 42: Daily Report .....	25
2.9.	Action 43: Jackpot Contributions On-Off Flag .....	28
2.10.	Action 44: Search for details of transactions (By House).....	30
2.11.	Action 45: Search for Jackpot Information .....	36
2.12.	Action 47: Get DemoToken .....	38
2.13.	Action 49: Query Game List .....	41
2.14.	Action 52: Query In-Game Player .....	43
2.15.	Action 54: Query Game Result.....	45
2.16.	Action 55: Check Cash Transfer (New) .....	47
2.17.	Action 58: Kick out Downline Users .....	49
2.18.	Action 61: Cash Transfer History (Including Rewards) .....	51
2.19.	Action 62: Check Event Ranking.....	53
2.20.	Action 64: Search for Old Details of Transactions in Games .....	55
2.21.	Action 65: Query Online Users With Balance > 0 .....	61
3.	Program Samples .....	63
3.1.	Java .....	63
3.2.	PHP .....	68
3.3.	C#.....	71
3.4.	Go Lang.....	75
3.5.	JavaScript.....	78
4.	FTP Functions .....	79
4.1.	Function Description .....	79
4.2.	Folder Path Description.....	79
4.3.	When create a new agent .....	79
5.	Appendix .....	80



5.1.	Error Codes.....	80
5.2.	Currency Code.....	82

# 1. Precautions and process description

## 1.1. Precautions

### ➤ Call-up method

#### ■ API URL

http:// {API\_Address}/apiRequest.do

#### ■ HTTP Method: POST

Parameter	Value
dc	Customer Domain Name
x	<ol style="list-style-type: none"> <li>1. Refer to Chapter “<a href="#">Functions</a>” to decide the function to call up</li> <li>2. Convert the parameters required according to the function into JSON String</li> <li>3. Use AES 128 bit to encrypt JSON String and base64 URL encode.</li> <li>4. For encryption process, please refer to “<a href="#">Program Samples</a>”.</li> </ol>

### ➤ Connection Description

- For the information about API address and encryption keys, please refer to “JDB API environment setting data.pdf”.
- To use this system, customers should provide the IP of the connected server.

### ➤ Return Format

- JSON format only
- **HTTP Compression:** User can choose whether to compress the return messages. If compression is needed, “Accept-Encoding”:gzip setting should be added to the header of “Http Request”.

### ➤ AES Encryption Notes

- Encrypted data is valid for 30 seconds. The data will expire if it is not sent in 30 seconds.  
**If the data expired, you will receive an error code of 9005. Please correct the system time.**
- The AES encryption process is a conversion supporting by fixed block size; therefore, in order to complete encryption successfully, the size of the data must be rounded up to a multiple of 16 bytes. If the size is insufficient, please fill up with blank space to the right.

## 1.2.Required function description

- Register account
  - Please refer to “[Action 12: Create Player](#)”
- Play the game
  - General player
    - ◆ Use the methods described in the “[Action 11: Obtain Token](#)” section to obtain the path.
    - ◆ Directly navigate to this path to log in to the system and play games.
  - Trial Player
    - ◆ Use the methods described in the “[Action 47: Get DemoToken](#)” section to obtain the path.
    - ◆ Directly navigate to this path to log in to the system and play games.
- Withdraw / Deposit
  - Please refer to “[Action 19: Withdraw / Deposit](#)”
  - You can verify that the serial No is successful by “[Action 55: Check Cash Transfer \(New\)](#)”.
- Check account balance
  - Please refer to “[Action 15: Search player information](#)”
- Query game history
  - Daily Report: Please refer to “[Action 42: Daily Report](#)”
  - Detailed transaction records:
    - ◆ Obtained through the API: Please refer to “[Action 29: Search for Details of Transactions in Games](#)”
    - ◆ Obtained through the FTP: Please refer to “[FTP Functions](#)”
    - ※ API only provide game history transaction information within 2 hours. For game history transaction information older than 2 hours, please use FTP.
    - ※ FTP provides trading information within 2 months
- Game result: Please refer to “[Action 54: Query Game Result](#)”

## 2. Functions

### 2.1.Action 11: Obtain Token

➤ Function Description

This function provides users to use the Token to login the system.

➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	11
<b>ts</b>	Long	Y	Current system time
<b>uid</b>	String(25)	Y	Player ID
<b>lang</b>	String(2)	N	Language en: English (Default) cn: Simplified Chinese th: Thai vn: Vietnamese ※Given any language that is not in the list above will be treated as English by default.
<b>gType</b>	String(2)	N	Game Type
<b>mType</b>	String(5)	N	Machine Type ※If you need to enter the game directly, you need to bring in mType and gType at the same time.
<b>remark</b>	String(100)	N	Remark
<b>windowMode</b>	String(1)	N	1: With using the JDB game lobby. (Default) ※ If gType and mType are not specified, users will directly enter the game lobby. ※ If gType and mType are specified, users will directly enter the game.  2: Without using the JDB game lobby. ※ "gType" and "mType" fields are required.
<b>isAPP</b>	Boolean	N	Whether to enter the game for the mobile app

			true: mobile app false: mobile webpage, computer webpage (Default)
<b>lobbyURL</b>	String(1000)	N	Game lobby URL <b>This parameter will only work when windowMode is 2.</b>
<b>moreGame</b>	Integer	N	0: Do not show more games 1: Show more games (default)
<b>mute</b>	Integer	N	Sound Setting: 0: Open (Default) 1: Mute
<b>cardGameGroup</b>	String(5)	N	Card Game play group <b>※Only a-z or A-Z or 0-9 allowed. (default:0)</b>
<b>isShowDollarSign</b>	Boolean	N	Whether to show dollar sign true: show dollar sign false: don't show dollar sign

Example:

```
{
  "action":11,
  "ts":1447490495704,
  "uid":"testpl01",
  "lang":"ch",
  "gType":"0",
  "mType":"8001",
  "windowMode":"2",
  "lobbyURL":"http://xxx.xxx.xxx/lobby.html"
}
```

## ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix
<b>err_text</b>	String(255)	Error message
<b>path</b>	String(255)	Login URL

## ➤ Operation Sample

## Http Request

```
http://{API_Address}/apiRequest.do?dc=jb1&x=y1wayxScczy6ABfElfaVwb5CdfHxBN_JGtxyBSvpo
hLQakP8M2QSetJNojJBzvrVChhq8YWWgBTdGeXFT4vSNQQ66epf4NqYeQ-
6bJ15pvHctQjQevulv7GHqOifuju_gY0H54FfVdQc9D5tKh1AZ-
9KDh9dwcGSszZxU7JOOoHdYITICO5LhvGmHxPBwZ
```

## Http Response (success)

```
{
  "status": "0000",
  "path": "https://xxx.xxx.xxx?x=Q0Y2NUZDNEFBODEzNUM5NURGRjc5QTUzRDk0RTIDRDzNE
Q0MjhEOEMzMSSSSSURDODBCNzdCMEE2OTE3MzcZrKE5RDNGQ0QzQjdFNTIBMje4Njg5ND
M4RDREMEI1NkYzQjkxOEu5QTU0QTM5QTY1MEUyNjYzMDg1RkFBQTM1OUU3RTI2RTBGQU
EzMUIOMkQ4NEFBjVDNkIFRkQzMzU1NzA5OUNDMkYxMzMzMyOTBNjBRURCNzExREMzRjh
DNzA1N0VFNEIzMzhCNTFEQ0FDODVEODQ3NEFDNTE1MEE1NTMzODQ="
}
```

## Http Response (error)

```
{
  "status": "9005",
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."
}
```

## 2.2.Action 12: Create Player

### ➤ Function Description

- Create a player (to create a Agent Level related account, please go to the background to create.)
- Role description: House → Manager → Master Agent → Agent → Player  
(cannot create an account across account level)
- **User ID under the same dc must be unique.**

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	12
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>uid</b>	String(25)	Y	Player ID Only a-z and 0-9 allowed. If the UID is in Uppercase, the system will automatically convert it to lowercase
<b>name</b>	String(50)	Y	Player Name Special symbols are not allowed: <>#"'%-+=*/ &
<b>credit_allocated</b>	Integer	N	Initial credits in the account.(By default is 0)

### Example:

```
{
  "action":12,
  "ts":1447452951820,
  "parent":"testag",
  "uid":"testpl01",
  "name":"testplayer",
  "credit_allocated":1000
}
```

➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix
<b>err_text</b>	String(255)	Error message

➤ Operation Sample

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-  
UWsfD6PHz9Ky-wM3mIC9EI4kwl_SIX3IpraVaCLnp-  
KOWsGkmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpl9jaj4XUgTK5yzb  
WKEddANjAAbxFOs=
```

Http Response (success)

```
{  
  "status": "0000"  
}
```

Http Response (error)

```
{  
  "status": "7602",  
  "err_text": "Account already exist. Please choose other user ID."  
}
```



## 2.3.Action 14: Player Status Management

### ➤ Function Description

- **Suspended:** Players can log in to the system but cannot play any games.  
If the player is playing in the game, there will appear the message “Your account has been suspended, please contact customer service.”, and then will divert to game hall or direct close game.
- **Locked:** Players cannot log in to the system.  
If the player is playing in the game, there will appear the message “Your account has been locked, please contact customer service.”, and then will direct logout from system.
- If a player is suspended or locked while in game, the game will immediate stopped and those pending transactions is counted.
- This function will immediate take effect once the player status been modified.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	14
<b>ts</b>	Long	Y	Currency system time
<b>parent</b>	String(19)	Y	Agent ID
<b>uid</b>	String(25)	Y	Player ID
<b>operation_code</b>	String(1)	Y	Account status 2: Suspended 3: Unsuspended 4: Locked 5: Unlocked

Example:

```
{
  "action":14,
  "ts":1447452951820,
  "parent":"testag",
  "uid":"testpl01",
  "operation_code":"2"
}
```

➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix
<b>err_text</b>	String(255)	Error message

➤ Operation Sample

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-  
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3IpraVaCLnp-  
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpl9jaj4XUgTK5yzb  
WKEddANjAAbxFOs=
```

Http Response (success)

```
{  
  "status": "0000"  
}
```

Http Response (error)

```
{  
  "status": "7502",  
  "err_text": "User is suspended."  
}
```

## 2.4.Action 15: Search player information

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	15
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>uid</b>	String(25)	Y	Player ID

Example:

```
{
  "action":15,
  "ts":1447452951820,
  "parent":"testag",
  "uid":"testpl01"
}
```

### ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See " <a href="#">Error Codes</a> " in Appendix
<b>err_text</b>	String(255)	Error message
<b>data</b>	JSON Array	
<b>uid</b>	String(25)	Player ID
<b>balance</b>	Number	Account balance
<b>parent</b>	String(19)	Agent ID
<b>username</b>	String(50)	Player Name
<b>currency</b>	String(3)	Currency types See " <a href="#">Currency Code</a> " in Appendix
<b>lvl</b>	Integer	Account types 0: Player
<b>locked</b>	Integer	Account status 0: Activate 1: Locked 2: Suspended 3: Suspended + Locked
<b>closed</b>	Integer	0: Account is enabled

		1: Account is closed
<b>jackpotFlag</b>	Integer	Whether with or without Jackpot (this setting is useful for those domain who have the Jackpot)  0: Enable  1: Disable

### ➤ Operation Sample

#### Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3lpraVaCLnp-
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpl9jaj4XUgTK5yzb
WKEddANjAAbxF0s=
```

#### Http Response (success)

```
{
  "status": "0000",
  "data": [{
    "uid": "testpl01",
    "balance": 901288.5,
    "parent": "testag",
    "username": "player01",
    "currency": "TB",
    "lvl": 0,
    "locked": 0,
    "closed": 0,
    "jackpotFlag": 0
  }]
}
```

#### Http Response (error)

```
{
  "status": "9005",
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."
}
```

## 2.5.Action 17: Kick Out Player

### ➤ Function Description

- If the player is online, player will receive a message of "You have been logged out! "
- If the player is playing the game, it will be terminated and settled directly.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	17
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>uid</b>	String(25)	Y	Player ID

Example:

```
{
  "action":17,
  "ts":1447452951820,
  "parent":"testag",
  "uid":"testpl01"
}
```

➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix
<b>err_text</b>	String(255)	Error message

➤ Operation Sample

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-  
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3IpraVaCLnp-  
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpl9jaj4XUgTK5yzb  
WKEddANjAAbxF0s=
```

Http Response (success)

```
{  
  "status": "0000"  
}
```

Http Response (error)

```
{  
  "status": "9005",  
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."  
}
```

## 2.6.Action 19: Withdraw / Deposit

### ➤ Function Description

- serialNo is a unique value.
- If there is no response when calling up API, please use the same serialNo when re-execution is needed to avoid duplicate execution.
- Error message 6005 means that duplicate execution occurs in this serialNo.
- If need to withdraw all out the player account balance, please use allCashOutFlag, which will cash out all the player account balance with decimals.
- Special Case Description: The following conditions will receive an error message 6901.
  - ◆ Player are not allowed to do any withdraw when they are playing in the **Card game** or staying in the Card Game Hall. Withdrawal are only allowed after **30 sec** they had leave the Card Game. Deposit are not affected.
  - ◆ Player are not allowed to do any withdraw when they are playing in the **Fish game** or staying in the Fish Game Hall. Withdrawal are only allowed after **5 sec** they had leave the Fish Game. Deposit are not affected.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	19
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>uid</b>	String(25)	Y	Player ID
<b>serialNo</b>	String(50)	Y	Transaction Serial Number, <b>English or digital only</b>
<b>allCashOutFlag</b>	String(1)	N	0: Not withdraw all (Default) 1: Withdraw all (include all balance with decimals)
<b>amount</b>	Number	N	Amount of withdrawal or deposit Positive: Deposit Negative: Withdrawal <b>※When allCashOutFlag is "0", this parameter need to be filled in.</b>
<b>remark</b>	String(15)	N	Remark

## Example:

```
{
  "action":19,
  "ts":1447452951820,
  "parent":"testag",
  "uid":"testpl01",
  "serialNo":"2016081200105",
  "allCashOutFlag":"0",
  "amount":1000,
  "remark":"test deposit"
}
```

## ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix
<b>userBalance</b>	Number	Total balance in the account
<b>userCashBalance</b>	Number	Cash balance in the account.
<b>agentCashBalance</b>	Number	Cash balance of the agent
<b>amount</b>	Number	The amount of money to withdraw/deposit
<b>serialNo</b>	String(50)	Transaction serial number entered
<b>pid</b>	Number	Transaction number
<b>payDate</b>	String(19)	Transaction date (dd-MM-yyyy HH:mm:ss)
<b>err_text</b>	String(255)	Error message

※ System has adjusted not to pre-store player balance for any games, therefore UserBalance and userCashBalance value will be same.

## ➤ Operation Sample

## Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-
UWsf6PHz9Ky-wM3mIC9EI4kwI_SIX3lpraVaCLnp-
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XUgTK5yzb
WKEddANjAAbxF0s=
```



**Http Response (success)**

```
{
  "status": "0000",
  "userBalance": 1000,
  "userCashBalance": 1000,
  "agentCashBalance": 500000,
  "amount": 1000,
  "serialNo": "2016081200105",
  "pid": 14425,
  "payDate": "08-12-2016 17:45:00"
}
```

**Http Response (error)**

```
{
  "status": "9005",
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."
}
```

## 2.7.Action 29: Search for Details of Transactions in Games

### ➤ Function Description

- Integrate details of game history transactions for all games and sort them by gType (game type).
- Search for a single parent ID's details of game history settled transaction with a given time range (search by lastModifyTime).
- The values of ss(seconds) in start time and end time must be 00.
- The maximum time range for each search is 15 minutes, and the display time of the latest information is 3 minutes from request time. For example, if you search request at 15:00:00, the latest information available is between 14:56:00 and 14:57:00.
- To obtain better response, specifying a time range less than 1 minute for each search is recommended.
- The function searches for the data according to the last modified time of the game record. Therefore, if the number of transactions found and transaction data on the win/loss statement are not matching, that is because the record has not cleared yet in the time range you searched.
- Provide game history transaction information within 2 hours.
- Each gambling game is considered as a transaction. When "hasGamble" is 1, it means that this is a gambling games transaction.

Example:

1. If a player bets 10 and wins 100 points in the game
    - ➔ bet = -10, gambleBet = 0, win = 100, total = 90, hasGamble = 0
  2. Same as above, if a player uses winning points to play gambling games and wins
    - ➔ bet = 0, gambleBet = -100, win = 200, total = 100, hasGamble = 1
- If the data is very huge, recommended to use the HTTP Compression to retrieve the data.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	29
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>starttime</b>	String(19)	Y	Start time of the searching range (dd-MM-yyyy HH:mm:ss)
<b>endtime</b>	String(19)	Y	End time of the searching range (dd-MM-yyyy HH:mm:ss)

Example:

```
{
  "action":29,
  "ts":1447452951820,
  "parent":"testag",
  "starttime":"08-12-2016 17:45:00",
  "endtime":"08-12-2016 17:46:00"
}
```

➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix ※No Data: 0000
<b>data</b>	JSON array	
<b>Betting records in slot games: gType = 0</b>		
<b>seqNo</b>	Long	Game sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Type
<b>mtype</b>	Integer	Machine Types(See “JDB_Games.pdf”)
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>gambleBet</b>	Double	Bet amount in Gambling games
<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(3)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
<b>jackpot</b>	Double	The amount of the jackpot won
<b>jackpotContribute</b>	Double	Jackpot contributions
<b>denom</b>	Double	Bet denomination
<b>lastModifyTime</b>	String(19)	Last modified time
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices.
<b>hasFreegame</b>	Integer	Free game 0: No 1: Yes
<b>hasGamble</b>	Integer	Gambling game 0: No 1: Yes
<b>systemTakeWin</b>	Integer	Mark this as the interruption if the game and it

		will be settled by the system 0: No 1: Yes
<b>Betting records in fishing game: gType = 7</b>		
seqNo	Long	Game sequence number.
playerId	String(25)	Player ID
gameDate	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
gType	Integer	Game Type
mtype	Integer	Machine Type
roomType	Integer	Game Room -1: Lobby (Achievement Game) 0:Small Bet 1:Medium sized bet 2:Large bet ※Real naming of game room differentiated by Machine Type
currency	String(3)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
bet	Double	Bet amount
win	Double	Winning points in the game
total	Double	Total Win Loss
denom	Double	Bet denomination
beforeBalance	Double	Balance before playing
afterBalance	Double	Balance after playing
lastModifyTime	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
playerIp	String(50)	Player login IP
clientType	String(20)	Players login from websites or mobile devices
<b>Betting records in Arcade games: gType = 9</b>		
seqNo	Long	Game sequence number
playerId	String(25)	Player ID
gType	Integer	Game Type
mtype	Integer	Machine Types(See “JDB_Games.pdf”)
gameDate	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
bet	Double	Bet amount
gambleBet	Double	Bet amount in Gambling games
win	Double	Winning points in the game
total	Double	Total Win Loss

<b>currency</b>	String(2)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
<b>denom</b>	Double	Bet denomination
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>hasBonusGame</b>	Integer	Bonus game 0: No 1: Yes
<b>hasGamble</b>	Integer	Gambling game 0: No 1: Yes
<b>Betting records in Lottery Game: gType = 12</b>		
<b>seqNo</b>	Long	Game sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Types
<b>mtype</b>	Integer	Machine Types(See “JDB_Games.pdf”)
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(3)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
<b>denom</b>	Double	Bet denomination
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>hasBonusGame</b>	Integer	Bonus game 0: No 1: Yes
<b>Betting records in Card Games: gType = 18</b>		
<b>seqNo</b>	Long	Game sequence number
<b>roundSeqNo</b>	Long	Game round sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Type
<b>mtype</b>	Integer	Machine Types(See “JDB_Games.pdf”)
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)

<b>bet</b>	Double	Bet amount
<b>win</b>	Double	Winning points in the game
<b>tax</b>	Double	Player pay winning tax in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(3)	Currency types See " <a href="#">Currency Code</a> " in Appendix
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>validBet</b>	Double	Valid Bet Amount
<b>err_text</b>	String(255)	Error message

### ➤ Operation Sample

#### Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3lpraVaCLnp-
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XUgTK5yzb
WKEddANjAAbxF0s=
```

#### Http Response (success)

```
{
  "status": "0000",
  "data": [
    {
      "seqNo": "5250145823902", "playerId": "test01", "gType": 0, "mtype": 8001, "gameDate": "06-06-2017 09:56:09", "bet": -0.4, "win": 0.4, "total": 0, "currency": "RB", "jackpot": 0, "jackpotContribute": 0.002, "denom": 0.02, "lastModifyTime": "06-06-2017 09:56:09", "playerIp": "10.20.6.86", "clientType": "WEB", "hasFreegame": 0, "hasGamble": 0, "gambleBet": 0, "systemTakeWin": 0},
    {
      "seqNo": "5250145823905", "playerId": "test01", "gType": 0, "mtype": 14001, "gameDate": "06-06-2017 09:56:30", "bet": 0, "win": 0, "total": -0.8, "currency": "RB", "jackpot": 0, "jackpotContribute": 0, "denom": 0.02, "lastModifyTime": "06-06-2017 09:56:30", "playerIp": "10.20.6.86", "clientType": "WEB", "hasFreegame": 0, "hasGamble": 1, "gambleBet": -0.8, "systemTakeWin": 0},
    {
      "seqNo": "43579741", "playerId": "test01", "gameDate": "07-06-2017 16:21:48", "gType": 7, "mtype": 7001, "roomType": 1, "currency": "RB", "bet": -62.5, "win": 19.75, "total": -
```

```

42.75,"denom":5,"beforeBalance":200095.6,"afterBalance":200052.85,"lastModifyTime":"
07-06-2017 16:21:55","playerIp":"10.20.6.86","clientType":"WEB"},
{"seqNo":5250152072553,"playerId":"sbtmopl01","gType":9,"mtype":9001,"gameDate":"25-07-2018
18:49:01","bet":-90,"gambleBet":0,"win":20,"total":-70,"currency":"RB","denom":1,"lastModifyTime":"25-07-
2018 18:49:06","playerIp":"10.20.9.250","clientType":"WEB","hasBonusGame":0,"hasGamble":0},
{"seqNo":5250152072557,"playerId":"sbtmopl01","gType":9,"mtype":9001,"gameDate":"25-07-2018
18:49:10","bet":0,"gambleBet":-20,"win":40,"total":20,"currency":"RB","denom":1,"lastModifyTime":"25-07-
2018 18:49:15","playerIp":"10.20.9.250","clientType":"WEB","hasBonusGame":0,"hasGamble":1},
{"seqNo":5250147282661,"playerId":"gh2alice1","gType":12,"mtype":12001,"gameDate":"14-05-2018
14:53:53","bet":-0.08,"win":0.05,"total":-0.03,"currency":"TB","denom":0.01,"lastModifyTime":"14-05-2018
14:53:53","playerIp":"10.20.9.250","clientType":"WEB","hasBonusGame":0},
{"seqNo":5250147282662,"playerId":"gh2alice1","gType":12,"mtype":12001,"gameDate":"14-05-2018
14:53:57","bet":-0.08,"win":0.05,"total":-0.03,"currency":"TB","denom":0.01,"lastModifyTime":"14-05-2018
14:53:57","playerIp":"10.20.9.250","clientType":"WEB","hasBonusGame":0},
{"seqNo":5250155819867,"roundSeqNo":5558,"playerId":"test888","gType":18,"mtype":18001,"gameDate":"1
3-07-2018 14:07:33","bet":-2,"win":6,"tax":-0.3,"total":5.7,"currency":"TB","lastModifyTime":"13-07-2018
14:07:33","playerIp":"10.20.9.250","clientType":"WEB","validBet":-2},
{"seqNo":5250155819872,"roundSeqNo":5559,"playerId":"test888","gType":18,"mtype":18001,"gameDate":"1
3-07-2018 14:07:47","bet":-8,"win":-8,"tax":0,"total":-8,"currency":"TB","lastModifyTime":"13-07-2018
14:07:47","playerIp":"10.20.9.250","clientType":"WEB","validBet":-8}
]
}

```

#### Http Response (error)

```

{
  "status":"9005",
  "err_text":"Assertion(SAML) didn't pass the timestamp validation."
}

```

#### Http Response (no data)

```

{
  "status":"0000",
  "data":[]
}

```

## 2.8.Action 42: Daily Report

### ➤ Function Description

- Search for a single agent ID's details of transactions in games in a given period of time
- If you specify the date 01-01-2016, you will obtain data in the range of 01-01-2016 12:00:00 to 01-02-2016 12:00:00.
- The latest information provided in each search is from the previous day.
  - ◆ Example 1: If the system time is 05-01-2016 18:23:00, you should specify a search date before 04-01-2016.
  - ◆ Example 2: If the system time is 05-01-2016 09:23:02, you should specify a search date before 03-01-2016(because the 04-01-2016 statement still in progress).
  - ◆ Example 3: If the system time is 05-01-2016 12:11:22, you should specify a search date before 03-01-2016.
- ( For data integrity, please retrieve information 1 hour later)
- Maximum time range for each search is 3 months.
- When the Card Game is enabled, all games are added with a "tax" return message.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	42
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>gType</b>	String(2)	Y	Game Type
<b>date</b>	String(10)	Y	Search date (dd-MM-yyyy)

Example:

```
{
  "action":42,
  "ts":1447452951820,
  "parent":"testag",
  "gType":"0",
  "date":"01-01-2016"
}
```



## ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix ※ No Data: 0000
<b>data</b>	JSON Object	
<b>uid</b>	String(25)	Player ID
<b>bet</b>	Double	Bet amount
<b>win</b>	Double	Winning points in the game
<b>netWin</b>	Double	Total Win Loss
<b>jackpot</b>	Double	The amount of the jackpot won
<b>jackpotContribute</b>	Double	Jackpot contributions
<b>count</b>	Integer	Bet Count
<b>validBet</b>	Double	Valid bet amount
<b>tax</b>	Double	Tax ※ Additional return column when Card Game is enabled.
<b>err_text</b>	String(255)	Error message

## ➤ Operation Sample

## Http Request

http://{API\_Address}/apiRequest.do?dc=jb&x=udbFVt\_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-UWsf6PHz9Ky-wM3mIC9EI4kwl\_SIX3lpraVaCLnp-KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw\_iU4554RuSsvHpI9jaj4XUgTK5yzbWKEddANjAAbxF0s=

## Http Response (success)

```
{
  "status": "0000",
  "data": [
    {
      "uid": "testpl01", "bet": -3760.8, "win": 2590.6, "netWin": -1170.2, "jackpot": 0, "jackpotContribute": -18.804, "count": 238, "validBet": -3760.8, "tax": -77.7,
    },
    {
      "uid": "testpl02", "bet": -2796.2, "win": 1789.65, "netWin": -1006.55, "jackpot": 0, "jackpotContribute": -13.981, "count": 274, "validBet": -2796.2, "tax": -55.66,
    },
    {
      "uid": "testpl03", "bet": -0.5, "win": 0.05, "netWin": -0.45, "jackpot": 0, "jackpotContribute": -
    }
  ]
}
```

```
0.0025,"count":1, "validBet ":-0.5, "tax ":-11.33}}
```

```
}
```

#### Http Response (error)

```
{
```

```
  "status":"9005",
```

```
  "err_text":"Assertion(SAML) didn't pass the timestamp validation."
```

```
}
```

#### Http Response (no data)

```
{
```

```
  "status":"0000",
```

```
  "data":[]
```

```
}
```

## 2.9.Action 43: Jackpot Contributions On-Off Flag

- Function Description  
TURN OFF Jackpot functions for some players.

- Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	43
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>uid</b>	String(25)	Y	Player ID
<b>jackpotFlag</b>	Integer	Y	0: On 1: Off

Example:

```
{  
  "action":43,  
  "ts":1447452951820,  
  "parent":"testag",  
  "uid":"testpl01",  
  "jackpotFlag":1  
}
```

➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix
<b>err_text</b>	String(255)	Error message

➤ Operation Sample

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-  
UWsfD6PHz9Ky-wM3mIC9EI4kwl_SIX3IpraVaCLnp-  
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpl9jaj4XUgTK5yzb  
WKEddANjAAbxF0s=
```

Http Response (success)

```
{  
  "status": "0000"  
}
```

Http Response (error)

```
{  
  "status": "9005",  
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."  
}
```

## 2.10. Action 44: Search for details of transactions (By House)

### ➤ Function Description

- Search for a single house ID's details of settled transaction in a given time range.  
(search by lastModifyTime)
- The values of ss(seconds) in start time and end time must be 00.
- The maximum time range for each search is 1 minutes, and the display time of the latest information is 3 minutes from request time.
- For example, if you search at 15:00:00, the latest information available is between 14:56:00 and 14:57:00.
- The function searches for the data according to the last modified time of the game record. Therefore, if the number of transactions found and transaction data on the win/loss statement are not matching, that is because the record has not cleared yet in the time range you searched.
- Provide game history transaction information within 2 hours. For transaction information older than 2 hours, please use FTP.
- Each gambling game is considered as a transaction. When "hasGamble" is 1, it means that this is a transaction in gambling games.

Example:

1. If a player bets 10 and wins 100 points in the game  
➔ bet = -10, gambleBet = 0, win = 100, total = 90, hasGamble = 0
2. Same as above, if a player uses winning points to play gambling games and wins  
➔ bet = 0, gambleBet = -100, win = 200, total = 100, hasGamble = 1

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	44
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	House ID
<b>starttime</b>	String(19)	Y	Start time of the searching range (dd-MM-yyyy HH:mm:ss)
<b>endtime</b>	String(19)	Y	End time of the searching range (dd-MM-yyyy HH:mm:ss)

Example:

```
{
  "action":44,
  "ts":1447452951820,
  "parent":"testhse",
  "starttime":"08-12-2016 17:45:00",
  "endtime":"08-12-2016 17:46:00"
}
```

➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix ※No Data: 0000
<b>data</b>	JSON Object	
<b>{parent}</b>	JSON Array	Agent ID
<b>Betting records in slot games: gType = 0</b>		
<b>seqNo</b>	Long	Game sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Type
<b>mtype</b>	Integer	Machine Types(See “JDB_Games.pdf”)
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>gambleBet</b>	Double	Bet amount in Gambling games
<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(3)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
<b>jackpot</b>	Double	The amount of the jackpot won
<b>jackpotContribute</b>	Double	Jackpot contributions
<b>denom</b>	Double	Bet denomination
<b>lastModifyTime</b>	String(19)	Last modified time
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices.
<b>hasFreeware</b>	Integer	Free game 0: No

		1: Yes
<b>hasGamble</b>	Integer	Gambling game 0: No 1: Yes
<b>systemTakeWin</b>	Integer	Mark this as the interruption if the game and it will be settled by the system 0: No 1: Yes
<b>Betting records in fishing game: gType = 7</b>		
<b>seqNo</b>	Long	Game sequence number.
<b>playerId</b>	String(25)	Player ID
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>gType</b>	Integer	Game Type
<b>mtype</b>	Integer	Machine Type
<b>roomType</b>	Integer	Game Room -1: Lobby (Achievement Game) 0:Small Bet 1:Medium sized bet 2:Large bet ※Real naming of game room differentiated by Machine Type
<b>currency</b>	String(3)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
<b>bet</b>	Double	Bet amount
<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>denom</b>	Double	Bet denomination
<b>beforeBalance</b>	Double	Balance before playing
<b>afterBalance</b>	Double	Balance after playing
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>Betting records in Arcade games: gType = 9</b>		
<b>seqNo</b>	Long	Game sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Type

<b>mtype</b>	Integer	Machine Types(See "JDB_Games.pdf")
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>gambleBet</b>	Double	Bet amount in Gambling games
<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(3)	Currency types See " <a href="#">Currency Code</a> " in Appendix
<b>denom</b>	Double	Bet denomination
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>hasBonusGame</b>	Integer	Bonus game 0: No 1: Yes
<b>hasGamble</b>	Integer	Gambling game 0: No 1: Yes

**Betting records in Lottery Game: gType = 12**

<b>seqNo</b>	Long	Game sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Types
<b>mtype</b>	Integer	Machine Types(See "JDB_Games.pdf")
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(3)	Currency types See " <a href="#">Currency Code</a> " in Appendix
<b>denom</b>	Double	Bet denomination
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>hasBonusGame</b>	Integer	Bonus game 0: No 1: Yes

**Betting records in Card Games: gType = 18**



<b>seqNo</b>	Long	Game sequence number
<b>roundSeqNo</b>	Long	Game round sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Type
<b>mtype</b>	Integer	Machine Types(See "JDB_Games.pdf")
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>win</b>	Double	Winning points in the game
<b>tax</b>	Double	Player pay winning tax in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(2)	Currency types See " <a href="#">Currency Code</a> " in Appendix
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>validBet</b>	Double	Valid Bet Amount
<b>err_text</b>	String(255)	Error message

### ➤ Operation Sample

#### Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3lpraVaCLnp-
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XUgTK5yzb
WKEddANjAAbxF0s=
```

#### Http Response (success)

```
{
  "status": "0000",
  "data": {
    "testag1": [
      {
        "seqNo": 5250568864735, "playerId": "testpl01", "gType": 0, "mtype": 8018, "gameDate": "03-05-2019 16:34:29", "bet": -0.05, "win": 0, "total": -0.05, "currency": "TB", "jackpot": 0, "jackpotContribute": -0.00025, "denom": 0.001, "lastModifyTime": "03-05-2019 16:34:29", "gameName": "", "playerIp": "10.20.9.250", "clientType": "WEB", "hasFreegame": 0, "hasGamble": 0, "gambleBet": 0, "systemTakeWin": 0, {"seqNo": 5250568864736, "playerId": "testpl01", "gType": 12, "mtype": 12001, "gameDate": "03-05-2019
```

```

16:34:50","bet":-0.008,"win":0.006,"total":-
0.002,"currency":"TB","denom":0.001,"lastModifyTime":"03-05-2019
16:34:50","playerIp":"10.20.9.250","clientType":"WEB","hasBonusGame":0}],
"testag2":[
{"seqNo":5250568864731,"playerId":"testpl01","gameDate":"03-05-2019
16:32:42","gType":7,"mtype":7003,"roomType":0,"gameName":"","currency":"TB","
bet":-0.22,"win":0,"total":-
0.22,"denom":1,"beforeBalance":111656.22,"afterBalance":111656,"lastModifyTim
e":"03-05-2019
16:32:42","playerIp":"10.20.9.250","clientType":"WEB"},{"seqNo":5250568864743,"
playerId":"testpl02","gType":9,"mtype":9001,"gameDate":"03-05-2019
16:36:36","bet":-1,"gambleBet":0,"win":0,"total":-
1,"currency":"TB","denom":0.1,"lastModifyTime":"03-05-2019
16:36:36","playerIp":"10.20.9.250","clientType":"WEB","hasBonusGame":0,"hasGa
mble":0},{"seqNo":5250568864737,"roundSeqNo":382342469,"playerId":"testpl02",
"gType":18,"mtype":18013,"gameDate":"03-05-2019 16:35:12","bet":-12,"win":-
12,"tax":0,"total":-12,"currency":"TB","lastModifyTime":"03-05-2019
16:35:17","playerIp":"10.20.9.250","clientType":"WEB","validBet":-12}]
}
}

```

#### Http Response (error)

```

{
  "status":"9005",
  "err_text":"Assertion(SAML) didn't pass the timestamp validation."
}

```

#### Http Response (no data)

```

{
  "status":"0000",
  "data":{"testag1":[],"testag2":[]}
}

```

## 2.11. Action 45: Search for Jackpot Information

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	45
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID

Example:

```
{
  "action":45,
  "ts":1447452951820,
  "parent":"testag"
}
```

### ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See " <a href="#">Error Codes</a> " in Appendix
<b>grand</b>	JSON Object	
<b>val</b>	Double	Current amount of the Grand Prize
<b>max</b>	Double	Maximum amount of the Grand Prize
<b>min</b>	Double	Minimum amount of the Grand Prize
<b>major</b>	JSON Object	
<b>val</b>	Double	Current amount of the Major Prize
<b>max</b>	Double	Maximum amount of the Major Prize
<b>min</b>	Double	Minimum amount of the Major Prize
<b>minor</b>	JSON Object	
<b>val</b>	Double	Current amount of the Minor Prize
<b>max</b>	Double	Maximum amount of the Minor Prize
<b>min</b>	Double	Minimum amount of the Minor Prize
<b>err_text</b>	String(255)	Error message

➤ Operation Sample

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-  
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3lpraVaCLnp-  
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpl9jaj4XUgTK5yzb  
WKEddANjAAbxF0s=
```

Http Response (success)

```
{  
  "status":"0000",  
  "grand":{"val":49269.20,"max":60000.00,"min":40000.00},  
  "major":{"val":4344.02,"max":6000.00,"min":4000.00},  
  "minor":{"val":390.59,"max":600.00,"min":300.00}  
}
```

Http Response (error)

```
{  
  "status":"9005",  
  "err_text":"Assertion(SAML) didn't pass the timestamp validation."  
}
```

## 2.12. Action 47: Get DemoToken

### ➤ Function Description

This function provides users the Token needed to login the system when playing trial game.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	47
<b>ts</b>	Long	Y	Current system time
<b>lang</b>	String(2)	N	Language en: English (Default) cn: Simplified Chinese th: Thai vn: Vietnamese ※Given any language that is not in the list above will be treated as English by default.
<b>gType</b>	String(2)	N	Game Type
<b>mType</b>	String(5)	N	Machine Type ※If you need to enter the game directly, you need to bring in gType and mType at the same time.
<b>windowMode</b>	String(1)	N	1: Use the JDB game lobby. (Default) ※ If gType and mType are not specified, users will directly enter the game lobby. ※ If gType and mType are specified, users will directly enter the game. 2: Don't use the JDB game lobby. ※ "gType" and "mType" fields are required.
<b>isAPP</b>	Boolean	N	Whether to enter the game for the mobile app true: mobile app false: mobile webpage, computer webpage (Default)
<b>lobbyURL</b>	String(1000)	N	Game lobby URL This parameter will only work when windowMode is 2.
<b>moreGame</b>	Integer	N	0: Do not show more games

			1: Show more games (default)
<b>mute</b>	Integer	N	Sound Setting: 0: Open (Default) 1: Mute
<b>isShowDollarSign</b>	Boolean	N	Whether to show dollar sign true: show dollar sign false: don't show dollar sign

Example:

```
{
  "action":47,
  "ts":1447490495704,
  "lang":"ch",
  "gType":"0",
  "mType":"8001"
  "windowMode":"2",
  "lobbyURL":"http://xxx.xxx.xxx/lobby.html"
}
```

#### ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See " <a href="#">Error Codes</a> " in Appendix
<b>err_text</b>	String(255)	Error message
<b>path</b>	String(255)	Login URL

#### ➤ Operation Sample

##### Http Request

http://{API\_Address}/apiRequest.do?dc=jb1&x=y1wayxScczy6ABfElfaVwb5CdfHxBN\_JGtxyBSvpo  
hLQakP8M2QSetJNojJBzvnrvChhq8YWWgBTdGeXFT4vSNQQ66epf4NqYeQ-  
6bJ15pvHctQjQevulv7GHqOifuju\_gY0H54FfVdQc9D5tKh1AZ-  
9KDh9dwcGSszZxU7JOOoHdYITICO5LhvGmHxPBwZ

##### Http Response (success)

```
{
  "status":"0000",
  "path":"https://xxx.xxx.xxx?x=Q0Y2NUZDNEFBODEzNUM5NURGRjc5QTUzRDk0RTIDRDUzNE  
Q0MjhEOEMzMURDODBCNzdCMEE2OTE3MzcZrKcE5RDNGQ0QzQjdFNTIBMjE4Njg5NDM4RD  
REMEI1NkYzQjkxOEU5QTU0QTM5QTY1MEUyNjYzMDg1RkFBQTM1OUU3RTI2RTBGQUEzMUI  
OMkQ4NEFBzjVDNkJFRkQzMzU1NzA5OUNDMkYxMzMzMyOTBNjBhBRURCNzExREMzRjhDNzA1  
NOVFNEIzMzhCNTFEQ0FDODVEODQ3NEFDNTE1MEE1NTMzODQ="
}
```

**Http Response (error)**

```
{
  "status": "9005",
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."
}
```

## 2.13. Action 49: Query Game List

### ➤ Function Description

- Get Game List of agent ID

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	49
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>lang</b>	String(2)	N	Language en: English (Default) ch: Simplified Chinese

Example:

```
{
  "action":49,
  "ts":1447452951820,
  "parent":"testag",
  "lang":"en"
}
```

### ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See " <a href="#">Error Codes</a> " in Appendix
<b>data</b>	JSON array	
<b>gType</b>	Integer	Game Type
<b>list</b>	JSON array	
<b>mType</b>	Integer	Machine Types
<b>isNew</b>	Boolean	Is New Game
<b>image</b>	String	Game Image URL, size is 250 x203. If need different size, please contact customer service
<b>name</b>	String	Game Name
<b>err_text</b>	String(255)	Error Message



➤ Operation Sample

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-  
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3IpraVaCLnp-  
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpl9jaj4XUgTK5yzb  
WKEddANjAAbxF0s=
```

Http Response (success)

```
{  
  "status": "0000",  
  "data": [  
    {  
      "gType": 0,  
      "list": [  
        {  
          "mType": 10,  
          "isNew": false,  
          "image": "https://xxx.xxx.xxx/images/xxx.jpg",  
          "name": "OpenSesame"  
        }  
      ]  
    },  
    {  
      "gType": 7,  
      "list": [  
        {  
          "mType": 7001,  
          "isNew": true,  
          "image": "https://xxx.xxx.xxx/images/xxx.jpg",  
          "name": "Dragon Treasure"  
        }  
      ]  
    },  
    {  
      "gType": 9,  
      "list": [  
        {  
          "mType": 9003,  
          "isNew": true,  
          "image": "https://xxx.xxx.xxx/images/xxx.jpg",  
          "name": "BirdsAndAnimals"  
        }  
      ]  
    }  
  ]  
}
```

Http Response (error)

```
{  
  "status": "9005",  
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."  
}
```

## 2.14. Action 52: Query In-Game Player

### ➤ Function Description

- Query is the player is inside the game playing, will response relevant information if player is in the game.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	52
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>uid</b>	String(25)	Y	Player ID

Example:

```
{
  "action":52,
  "ts":1447452951820,
  "parent":"testag",
  "uid":" testpl01"
}
```

### ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See " <a href="#">Error Codes</a> " in Appendix
<b>data</b>	JSON array	
<b>gType</b>	Integer	Game Type
<b>mType</b>	Integer	Machine Types
<b>loginFrom</b>	String(20)	Player login from websites or mobile devices
<b>ipAddr</b>	String(50)	Player login IP
<b>loginTime</b>	String(19)	Login game time (dd-MM-yyyy HH:mm:ss)
<b>balance</b>	Number	Balance
<b>err_text</b>	String(255)	Error message

➤ Operation Sample

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-  
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3lpraVaCLnp-  
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XUgTK5yzb  
WKEddANjAAbxF0s=
```

Http Response (success)

```
{  
  "status": "0000",  
  "data": {  
    "gType": 0,  
    "mType": 8011,  
    "loginFrom": "WEB",  
    "ipAddr": "10.20.100.13",  
    "loginTime": "06-11-2017 16:49:18",  
    "balance": 1000  
  }  
}
```

Http Response (error)

```
{  
  "status": "7505",  
  "err_text": "User is not playing"  
}
```

## 2.15. Action 54: Query Game Result

### ➤ Function Description

- Query the player game results
- Query without gType and seqNo: Report main page will be displayed.
- Query with gType and seqNo: Detailed game result of seqNo will be displayed.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	54
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>uid</b>	String(25)	Y	Player ID
<b>lang</b>	String(5)	N	Language en: English (Default) ch: Simplified Chinese
<b>gType</b>	Integer	N	Game Type(See "JDB_Games.pdf") If "gType" is specified, "seqNo" needed to specified as well
<b>seqNo</b>	Long	N	Game sequence number If "seqNo" is specified, "gType" needed to specified as well
<b>showUid</b>	Integer	N	Whether show player id 0: not show id 1: show id(default)

### Example:

```
{
  "action":54,
  "ts":1447452951820,
  "parent":"testag",
  "uid":" testpl01",
  "gType": 0,
  "seqNo":1234567890
}
```

## ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix
<b>data</b>	JSON array	
<b>path</b>	String	Error message
<b>err_text</b>	String(255)	Login URL

## ➤ Operation Sample

## Http Request

http://{API\_Address}/apiRequest.do?dc=jb1&x=y1wayxScczy6ABfElfaVwb5CdfHxBN\_JGtxyBSvpo  
hLQakP8M2QSetJNojJBzvnrvChhq8YWWgBTdGeXFTH4vSNQQ66epf4NqYeQ-  
6bJ15pvHctQjQevulv7GHqOifuju\_gY0H54FfVdQc9D5tKh1AZ-  
9KDh9dwcGSszZxU7JOOoHdYITICO5LhvGmHxPBwZ

## Http Response (success)

```
{
  "status": "0000",
  "data": [
    {
      "path": "https://xxx.xxx.xxx/...."
    }
  ]
}
```

## Http Response (error)

```
{
  "status": "9005",
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."
}
```

## 2.16. Action 55: Check Cash Transfer (New)

### ➤ Function Description

- Search for a certain user's cash transfer history in a given time range.
- Maximum time range for each search is 3 months.
- It will return error code 9015 when search result is no data.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	55
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>serialNo</b>	String(50)	Y	User transaction serial number User transaction serial number defined in "Withdraw/Deposit"

Example:

```
{
  "action":55,
  "ts":1447452951820,
  "parent":"testag",
  "serialNo":"2016081200105"
}
```

### ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See " <a href="#">Error Codes</a> " in Appendix ✖No Data: 9015
<b>data</b>	JSON array	
<b>uid</b>	String(25)	Player ID
<b>action</b>	String(10)	withdraw / deposit / reward
<b>amount</b>	Number	Transfer Amount
<b>plsBalance</b>	Number	Account Balance before Transfer
<b>date</b>	String(19)	Transaction date (dd-MM-yyyy HH:mm:ss)
<b>pid</b>	Integer	Transaction number
<b>remark</b>	String(15)	Remark

<b>afterBalance</b>	Number	Balance after deposit / withdraw
<b>err_text</b>	String(255)	Error message

➤ Operation Sample

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3IpraVaCLnp-
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XUgTK5yzb
WKEddANjAAbxF0s=
```

Http Response (success)

```
{
  "status": "0000",
  "data": [{
    "uid": "testpl01",
    "action": "deposit",
    "amount": 100,
    "plsBalance": 998800,
    "date": "13-02-2015 13:04:22",
    "pid": 11491,
    "remark": "test",
    "afterBalance": 9988900
  }]
}
```

Http Response (error)

```
{
  "status": "9015",
  "err_text": "Data does not exist."
}
```

## 2.17. Action 58: Kick out Downline Users

### ➤ Function Description

- This action will work for all downline users of the parent.
- If the player is online, he/she will receive a message.
- If the player is playing the game, it will be terminated and settled directly.
- If this action is working in process, it will return error code 9017
- ※ You can use the searching services provided by “Search for Details of Transactions in Games” to search for the results.
- ※ Example  
 If the parent is house account, the all downline users of the house account will be kicked out. (Include manager, master agent, agent and player)  
 If the parent is agent account, the all players of the agent account will be kicked out

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	58
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Parent ID

Example:

```
{
  "action":58,
  "ts":1447452951820,
  "parent":"testag"
}
```



➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix
<b>err_text</b>	String(255)	Error message

➤ Operation Sample

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-  
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3IpraVaCLnp-  
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpl9jaj4XUgTK5yzb  
WKEddANjAAbxF0s=
```

Http Response (success)

```
{  
  "status": "0000"  
}
```

Http Response (error)

```
{  
  "status": "9005",  
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."  
}
```

## 2.18. Action 61: Cash Transfer History (Including Rewards)

### ➤ Function Description

- Search for a certain user's cash transfer & rewards history in a given time range.
- Maximum time range for each search is 3 months.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	61
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>uid</b>	String(25)	Y	Player ID
<b>starttime</b>	String(19)	Y	Start Time of the searching range. (dd-MM-yyyy HH:mm:ss)
<b>endtime</b>	String(19)	Y	End Time of the searching range. (dd-MM-yyyy HH:mm:ss)

Example:

```
{
  "action":61,
  "ts":1447452951820,
  "parent":"testag",
  "uid":"testpl01",
  "starttime":"08-12-2016 17:45:00",
  "endtime":"08-12-2016 18:00:00"
}
```

### ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See " <a href="#">Error Codes</a> " in Appendix ※No Data: 0000
<b>data</b>	JSON array	
<b>pid</b>	Integer	Transaction Number
<b>payDate</b>	String(19)	Transaction Date. (dd-MM-yyyy HH:mm:ss)
<b>action</b>	String(10)	Withdraw / Deposit / Rewards
<b>amount</b>	Number	Transaction Amount

<b>balance</b>	Number	Account Balance
<b>serialNo</b>	String(50)	Transaction Serial No
<b>remark</b>	String(100)	Remark
<b>err_text</b>	String(255)	Error Message

## ➤ Operation Sample

## Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3IpraVaCLnp-
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpl9jaj4XUgTK5yzb
WKEddANjAAbxF0s=
```

## Http Response (success)

```
{
  "status":"0000",
  "data":[
    {"pid":4934,"payDate":"30-08-2018 14:29:39","action":"deposit","amount":100,"balance":100,
    "remark":"","serialNo":"2018083000105"},
    {"pid":4938,"payDate":"30-08-2018 14:35:29","action":"withdraw","amount":50,"balance":50,
    "remark":"","serialNo":"2018083000106"},
    {"pid":4950,"payDate":"30-08-2018 14:58:34","action":"reward","amount":1000,"balance":1050,
    "remark":"2018-08-10 财神发发发","serialNo":"20180830_123456_1_1"}
  ]
}
```

## Http Response (error)

```
{
  "status":"9005",
  "err_text":"Assertion(SAML) didn't pass the timestamp validation."
}
```

## Http Response (no data)

```
{
  "status":"0000",
  "data":[]
}
```

## 2.19. Action 62: Check Event Ranking

### ➤ Function Description

- Query the ranking according to event date range.

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	62
<b>ts</b>	Long	Y	Current system time
<b>startDate</b>	String(10)	Y	Start Time of the searching range. (dd-MM-yyyy)
<b>endDate</b>	String(10)	Y	End Time of the searching range. (dd-MM-yyyy)

Example:

```
{
  "action":62,
  "ts":1447452951820,
  "startDate":"29-08-2018",
  "endDate":"29-08-2018"
}
```

### ➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix ✖ No Data: 9015
<b>data</b>	JSON array	
<b>activityDate</b>	String(10)	Event Date (dd-mm-yyyy)
<b>activityNo</b>	String(20)	Event Code
<b>activityName</b>	String(100)	Event Name
<b>awardData</b>	JSON array	
<b>awardId</b>	String(3)	Reward Code
<b>awardName</b>	String(50)	Reward Name
<b>rankData</b>	JSON array	
<b>rank</b>	Int	Reward Ranking
<b>uid</b>	String(25)	Player ID

<b>amount</b>	Double	Reward
<b>parent</b>	String(19)	Agent ID
<b>err_text</b>	String(255)	Error Messages

### ➤ Operation Sample

#### Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3lpraVaCLnp-
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XUgTK5yzb
WKEddANjAAbxF0s=
```

#### Http Response (success)

```
{
  "status": "0000",
  "data": [
    {
      "activityNo": "1535335791346",
      "activityName": "财神发发发",
      "activityDate": "29-08-2018",
      "awardData": [
        {
          "awardId": "1",
          "awardName": "白银",
          "rankData": [
            {"rank": 3, "amount": 500, "id": "testpl01", "parent": "testag"},
            {"rank": 4, "amount": 500, "uid": "testpl02", "parent": "testag"},
            {"rank": 7, "amount": 300, "uid": "testpl03", "parent": "testag"}
          ]
        }, {
          "awardId": "2",
          "awardName": "黄金",
          "rankData": [
            {"rank": 2, "amount": 10000, "uid": "testpl04", "parent": "testag"}
          ]
        }
      ]
    }
  ]
}
```

#### Http Response (error)

```
{
  "status": "9005",
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."
}
```

## 2.20. Action 64: Search for Old Details of Transactions in Games

### ➤ Function Description

- Provide data from 2021/01/07 (UTC +08:00).
- Integrate details of game history transactions for all games and sort them by gType(game type).
- Search for a single parent ID's details of game history settled transaction with a given time range (search by lastModifyTime).
- The values of ss(seconds) in start time and end time must be 00.
- The maximum time range for each search is 5 minutes.
- To obtain better response when huge data and heavy traffic, specifying a time range less than 1 minute for each search is recommended.
- The function searches for the data according to the last modified time of the game record. Therefore, if the number of transactions found and transaction data on the win/loss statement are not matching, that is because the record has not cleared yet in the time range you searched.
- Provide game history transaction information between 2 hours and 60 days.
- Each gambling game is considered as a transaction. When "hasGamble" is 1, it means that this is a gambling games transaction.

Example:

2. If a player bets 10 and wins 100 points in the game
  - ➔ bet = -10, gambleBet = 0, win = 100, total = 90, hasGamble = 0
2. Same as above, if a player uses winning points to play gambling games and wins
  - ➔ bet = 0, gambleBet = -100, win = 200, total = 100, hasGamble = 1

### ➤ Parameter

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	64
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID
<b>starttime</b>	String(19)	Y	Start time of the searching range (dd-MM-yyyy HH:mm:ss)
<b>endtime</b>	String(19)	Y	End time of the searching range (dd-MM-yyyy HH:mm:ss)

Example:

```
{
  "action":64,
  "ts":1447452951820,
  "parent":"testag",
  "starttime":"08-12-2016 17:45:00",
  "endtime":"08-12-2016 17:46:00"
}
```

➤ Return Results

Parameter	Format	Description
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix ※No Data: 0000
<b>data</b>	JSON array	
<b>Betting records in slot games: gType = 0</b>		
<b>seqNo</b>	Long	Game sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Type
<b>mtype</b>	Integer	Machine Types(See “JDB_Games.pdf”)
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>gambleBet</b>	Double	Bet amount in Gambling games
<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(3)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
<b>jackpot</b>	Double	The amount of the jackpot won
<b>jackpotContribute</b>	Double	Jackpot contributions
<b>denom</b>	Double	Bet denomination
<b>lastModifyTime</b>	String(19)	Last modified time
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices.
<b>hasFreegame</b>	Integer	Free game 0: No 1: Yes
<b>hasGamble</b>	Integer	Gambling game 0: No

		1: Yes
<b>systemTakeWin</b>	Integer	Mark this as the interruption if the game and it will be settled by the system 0: No 1: Yes
<b>Betting records in fishing game: gType = 7</b>		
<b>seqNo</b>	Long	Game sequence number.
<b>playerId</b>	String(25)	Player ID
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>gType</b>	Integer	Game Type
<b>mtype</b>	Integer	Machine Type
<b>roomType</b>	Integer	Game Room -1: Lobby (Achievement Game) 0:Small Bet 1:Medium sized bet 2:Large bet ※Real naming of game room differentiated by Machine Type
<b>currency</b>	String(3)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
<b>bet</b>	Double	Bet amount
<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>denom</b>	Double	Bet denomination
<b>beforeBalance</b>	Double	Balance before playing
<b>afterBalance</b>	Double	Balance after playing
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>Betting records in Arcade games: gType = 9</b>		
<b>seqNo</b>	Long	Game sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Type
<b>mtype</b>	Integer	Machine Types(See “JDB_Games.pdf”)
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>gambleBet</b>	Double	Bet amount in Gambling games



<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(2)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
<b>denom</b>	Double	Bet denomination
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>hasBonusGame</b>	Integer	Bonus game 0: No 1: Yes
<b>hasGamble</b>	Integer	Gambling game 0: No 1: Yes
<b>Betting records in Lottery Game: gType = 12</b>		
<b>seqNo</b>	Long	Game sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Types
<b>mtype</b>	Integer	Machine Types(See “JDB_Games.pdf”)
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>win</b>	Double	Winning points in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(3)	Currency types See “ <a href="#">Currency Code</a> ” in Appendix
<b>denom</b>	Double	Bet denomination
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>hasBonusGame</b>	Integer	Bonus game 0: No 1: Yes
<b>Betting records in Card Games: gType = 18</b>		
<b>seqNo</b>	Long	Game sequence number
<b>roundSeqNo</b>	Long	Game round sequence number
<b>playerId</b>	String(25)	Player ID
<b>gType</b>	Integer	Game Type

<b>mtype</b>	Integer	Machine Types(See "JDB_Games.pdf")
<b>gameDate</b>	String(19)	Date of playing. (dd-MM-yyyy HH:mm:ss)
<b>bet</b>	Double	Bet amount
<b>win</b>	Double	Winning points in the game
<b>tax</b>	Double	Player pay winning tax in the game
<b>total</b>	Double	Total Win Loss
<b>currency</b>	String(3)	Currency types See " <a href="#">Currency Code</a> " in Appendix
<b>lastModifyTime</b>	String(19)	Last modified time. (dd-MM-yyyy HH:mm:ss)
<b>playerIp</b>	String(50)	Player login IP
<b>clientType</b>	String(20)	Players login from websites or mobile devices
<b>validBet</b>	Double	Valid Bet Amount
<b>err_text</b>	String(255)	Error message

### ➤ Operation Sample

#### Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQO-
UWsf6PHz9Ky-wM3mIC9EI4kwl_SIX3lpraVaCLnp-
KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XUgTK5yzb
WKEddANjAAbxF0s=
```

#### Http Response (success)

```
{
  "status":"0000",
  "data":[
    {"seqNo":5250145823902,"playerId":"test01","gType":0,"mtype":8001,"gameDate":"06-
    06-2017 09:56:09","bet":-
    0.4,"win":0.4,"total":0,"currency":"RB","jackpot":0,"jackpotContribute":-
    0.002,"denom":0.02,"lastModifyTime":"06-06-2017
    09:56:09","playerIp":"10.20.6.86","clientType":"WEB","hasFreegame":0,"hasGamble":0,"g
    ambleBet":0,"systemTakeWin":0},
    {"seqNo":5250145823905,"playerId":"test01","gType":0,"mtype":14001,"gameDate":"06-
    06-2017 09:56:30","bet":0,"win":0,"total":-
    0.8,"currency":"RB","jackpot":0,"jackpotContribute":0,"denom":0.02,"lastModifyTime":"0
    6-06-2017
    09:56:30","playerIp":"10.20.6.86","clientType":"WEB","hasFreegame":0,"hasGamble":1,"g
    ambleBet":-0.8,"systemTakeWin":0},
    {"seqNo":43579741,"playerId":"test01","gameDate":"07-06-2017
```

```

16:21:48", "gType":7, "mtype":7001, "roomType":1, "currency": "RB", "bet": -
62.5, "win":19.75, "total": -
42.75, "denom":5, "beforeBalance":200095.6, "afterBalance":200052.85, "lastModifyTime": "
07-06-2017 16:21:55", "playerIp": "10.20.6.86", "clientType": "WEB"},
{"seqNo":5250152072553, "playerId": "sbtmppy01", "gType":9, "mtype":9001, "gameDate": "25-07-2018
18:49:01", "bet": -90, "gambleBet":0, "win":20, "total": -70, "currency": "RB", "denom":1, "lastModifyTime": "25-07-
2018 18:49:06", "playerIp": "10.20.9.250", "clientType": "WEB", "hasBonusGame":0, "hasGamble":0},
{"seqNo":5250152072557, "playerId": "sbtmppy01", "gType":9, "mtype":9001, "gameDate": "25-07-2018
18:49:10", "bet":0, "gambleBet": -20, "win":40, "total":20, "currency": "RB", "denom":1, "lastModifyTime": "25-07-
2018 18:49:15", "playerIp": "10.20.9.250", "clientType": "WEB", "hasBonusGame":0, "hasGamble":1},
{"seqNo":5250147282661, "playerId": "gh2alice1", "gType":12, "mtype":12001, "gameDate": "14-05-2018
14:53:53", "bet": -0.08, "win":0.05, "total": -0.03, "currency": "TB", "denom":0.01, "lastModifyTime": "14-05-2018
14:53:53", "playerIp": "10.20.9.250", "clientType": "WEB", "hasBonusGame":0},
{"seqNo":5250147282662, "playerId": "gh2alice1", "gType":12, "mtype":12001, "gameDate": "14-05-2018
14:53:57", "bet": -0.08, "win":0.05, "total": -0.03, "currency": "TB", "denom":0.01, "lastModifyTime": "14-05-2018
14:53:57", "playerIp": "10.20.9.250", "clientType": "WEB", "hasBonusGame":0},
{"seqNo":5250155819867, "roundSeqNo":5558, "playerId": "test888", "gType":18, "mtype":18001, "gameDate": "1
3-07-2018 14:07:33", "bet": -2, "win":6, "tax": -0.3, "total":5.7, "currency": "TB", "lastModifyTime": "13-07-2018
14:07:33", "playerIp": "10.20.9.250", "clientType": "WEB", "validBet": -2},
{"seqNo":5250155819872, "roundSeqNo":5559, "playerId": "test888", "gType":18, "mtype":18001, "gameDate": "1
3-07-2018 14:07:47", "bet": -8, "win": -8, "tax":0, "total": -8, "currency": "TB", "lastModifyTime": "13-07-2018
14:07:47", "playerIp": "10.20.9.250", "clientType": "WEB", "validBet": -8}
]
}

```

#### Http Response (error)

```

{
  "status": "9005",
  "err_text": "Assertion(SAML) didn't pass the timestamp validation."
}

```

#### Http Response (no data)

```

{
  "status": "0000",
  "data": []
}

```

## 2.21. Action 65: Query Online Users With Balance > 0

### ➤ Function Description

- Query online users with balance > 0

Parameter	Format	Mandatory	Description
<b>action</b>	Integer	Y	65
<b>ts</b>	Long	Y	Current system time
<b>parent</b>	String(19)	Y	Agent ID

Example:

```
{
  "action":65,
  "ts":1447452951820,
  "parent":"testag"
}
```

### ➤ Return Results

Parameter	Format	Discription
<b>status</b>	String(4)	Success: 0000 Error: See “ <a href="#">Error Codes</a> ” in Appendix ✖ No Data: 0000
<b>data</b>	JSON array	
<b>uid</b>	String(25)	Player ID
<b>balance</b>	Number	Account balance

### ➤ Example

#### Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-UWsfd6PHz9Ky-wM3mIC9E14kw1_SlX3lpraVaCLnp-K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XUgTK5yzbwKEddANjAAbxF0s=
```

#### Http Response (success)

```
{"status":"0000","data":[{"uid":"testplayer","balance":7101.49}]}
```

#### Http Response (error)

```
{
  "status":"9005",
  "err_text":"Assertion(SAML) didn't pass the timestamp validation."
}
```

Http Response (no data)

```
{"status": "0000", "data": []}
```

## 3. Program Samples

### ➤ Description:

- Get token for player login game for example.
- This sample consists of three main parts:
  - ◆ AES Encryption: Give the data required to set “request” and iv, key and then encrypt using AES encryption with CBC modes (NoPadding).
  - ◆ Encrypt and Send: Give the data required to set “request”, activate the encryption process described above, and send the request to the server after base64 URL encode.
  - ◆ Program Testing: Set the data required to set “request”, activate the encryption process described above, send the programs, and receive the results.

### 3.1.Java

#### ➤ AES Encryption

```
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.net.util.Base64;

public class MEncrypt {
    public static String encrypt(String data, String key, String iv) throws Exception {
        Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding");
        int blockSize = cipher.getBlockSize();
        byte[] dataBytes = data.getBytes("UTF-8");
        int plainTextLength = dataBytes.length;
        if (plainTextLength % blockSize != 0) {
            plainTextLength = plainTextLength + (blockSize - plainTextLength % blockSize);
        }
        byte[] plaintext = new byte[plainTextLength];
        System.arraycopy(dataBytes, 0, plaintext, 0, dataBytes.length);
        SecretKeySpec keyspec = new SecretKeySpec(key.getBytes(), "AES");
        IvParameterSpec ivspec = new IvParameterSpec(iv.getBytes());
        cipher.init(Cipher.ENCRYPT_MODE, keyspec, ivspec);
        byte[] encrypted = cipher.doFinal(plaintext);
        return Base64.encodeBase64URLSafeString(encrypted);
    }
}
```

➤ Encrypt and Send

```
import java.util.ArrayList;
import java.util.List;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClientBuilder;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
import com.google.gson.JsonObject;
import com.google.gson.JsonParser;

public class sample {

    public static JsonObject jsonObjectencryptAndQuery(JsonObject v, boolean zipResponse)
throws Exception {

        JsonObject rJson = null;
        String x;

        // get a client
        CloseableHttpClient demo = HttpClientBuilder.create().build();
        String vString = v.toString();

        // encrypt
        x = encrypt(vString, $ { AES_KEY }, $ { AES_IV });
```

```
// build request
List<NameValuePair> paramList = new ArrayList<>();
paramList.add(new BasicNameValuePair("dc",  ${ DC }));
paramList.add(new BasicNameValuePair("x", x));
UrlEncodedFormEntity entity = new UrlEncodedFormEntity(paramList, "UTF-8");
HttpPost httpPost = new HttpPost(${ API URL });
httpPost.setEntity(entity);
if (zipResponse) {
    httpPost.setHeader("Accept-Encoding", "gzip");
}

HttpEntity httpEntity = null;
try {
    // Resolve response
    HttpResponse response = demo.execute(httpPost);
    httpEntity = response.getEntity();
    String responseEntityString = EntityUtils.toString(httpEntity);
    System.out.println(responseEntityString);
    JsonParser parser = new JsonParser();
    rJson = parser.parse(responseEntityString).getAsJsonObject();
} finally {
    EntityUtils.consume(httpEntity);
}
return rJson;
}
```



➤ Sample – Action 11

```
@Test
public void testPerformAction11() throws Exception {
    // prepare data
    JsonObject v = generateNewVector();
    v.addProperty("action", 11);
    v.addProperty("ts", $ {system time});
    v.addProperty("uid", $ {player id});
    v.addProperty("lang", "en");
    v.addProperty("gType", "0");
    v.addProperty("mType", "8001");
    v.addProperty("windowMode", "2");

    // make request.
    result = encryptAndQuery(v, false);
    if (result == null) {
        // Do Error handlings here
    }

    // get status and return
    status = result.get("status").getAsString();
    Assert.assertTrue(status.startsWith("0000"));
}
```

➤ Sample – Action 12

```
@Test
public void testPerformAction12() throws Exception {
    // prepare data
    JsonObject v = generateNewVector();
    v.addProperty("action", 12);
    v.addProperty("ts", ${system time});
    v.addProperty("parent", ${parent});
    v.addProperty("uid", ${uid});
    v.addProperty("name", ${name});
    v.addProperty("credit_allocated", ${credit allocated});

    // make request.
    result = encryptAndQuery(v, false);
    if (result == null) {
        // Do Error handlings here
    }

    // get status and return
    status = result.get("status").getString();
    Assert.assertTrue(status.startsWith("0000"));
}
```

## 3.2.PHP

### ➤ AES Encryption and Decryption (MCrypt.php)

```
<?php
class MCrypt
{
    private $key = ""; // change it
    private $iv = ""; // change it

    function __construct(){}

    function encrypt($str){
        $key = $this->key;
        $iv = $this->iv;
        $str= $this->padString($str);
        $encrypted = openssl_encrypt($str, 'AES-128-CBC', $key, OPENSSL_RAW_DATA, $iv);
        $data = base64_encode($encrypted);
        $data = str_replace(array('+','/','='),array('-','_',''),$data);
        return $data;
    }

    function decrypt($code) {
        $code = str_replace(array('-', '_'), array('+', '/'), $code);
        $code = base64_decode($code);
        $key = $this->key;
        $iv = $this->iv;
        $decrypted = openssl_decrypt($code, 'AES-128-CBC', $key, OPENSSL_NO_PADDING, $iv);
        return utf8_encode(trim($decrypted));
    }

    private function padString($source) {
        $paddingChar = ' ';
        $size = 16;
        $x = strlen($source) % $size;
        $padLength = $size - $x;
        for ($i = 0; $i < $padLength; $i++) {
            $source .= $paddingChar;
        }
        return $source;
    }
}
?>
```

### ➤ Sample – Action 12

```
<?php
    include 'MCrypt.php';

    // Action 12:required parameter information
    $mc = new MCrypt();
    $now = round(microtime(true)*1000);
    $dc = ${DC};
    $parent = 'agent';
    $uid = 'playerId';
    $jsonString =
'{"action":12,"ts":'.$now.',"parent":'.$parent.',"uid":'.$uid.',"name":"playerName","c
redit_allocated":0}';

    // encrypt data
    $encryptData = $mc->encrypt($jsonString);
    $data = array('dc'=>$dc, 'x'=>$encryptData);

    // check environment Settings file
    $apiurl = ${API URL};

    // post
    $ch = curl_init($apiurl);
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($data));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);

    //parse json
    $json = json_decode($response, true);

    // this is a one-time use only token
    var_dump($json);

?>
```

## ➤ Sample – Action 11

```
<?php
    include 'MCrypt.php';

    // Action 11:required parameter information
    $mc = new MCrypt();
    $now = round(microtime(true)*1000);
    $dc = ${DC};
    $uid = 'playerId';
    $jsonString = '{"action":11, "ts":'.$now.', "uid": "'.$uid.'" , "lang":"ch", "gType":"0",
    "mType":"8001", "windowMode":2}';

    // encrypt data
    $encryptData = $mc->encrypt($jsonString);
    $data = array('dc'=>$dc, 'x'=>$encryptData);

    // check environment Settings file
    $apiurl = ${API URL};

    // post
    $ch = curl_init($apiurl);
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($data));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);

    //parse json
    $json = json_decode($response, true);
    $url = $json['path'];

    echo "<script type='text/javascript'>";
    echo "window.location.href='$url'";
    echo "</script>";

?>
```

### 3.3.C#

➤ Sample – Creating user and logging in system

```
using System;
using System.Security.Cryptography;
using System.Text;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using RestSharp;

public class AESProvider{
    private static readonly string AES_KEY = ${AES KEY}; // change it
    private static readonly string AES_IV = ${AES IV}; // change it
    private static readonly char[] padding = { '=' };

    public static void Main(string[] args) {
        // Action 11:required parameter information
        String dc = ${DC};
        String uid = ${uid};
        String jsonString = "{\"action\":\"11\", \"ts\":\"" +
        DateTimeOffset.UtcNow.ToUnixTimeMilliseconds() + ", \"uid\": " + uid + ", \"lang\":\"ch\",
        \"gType\":\"0\", \"mType\":\"8001\", \"windowMode\":\"2\"}";

        // encrypt data
        String x = AESEncryptToString(jsonString);

        // check environment Settings file
        String apiurl = ${API URL};
        var client = new RestClient(apiurl);

        // post
        var request = new RestRequest(Method.POST);
        request.AddHeader("cache-control", "no-cache");
        request.AddHeader("content-type", "application/x-www-form-urlencoded");
        request.AddParameter("application/x-www-form-urlencoded", "dc=" + dc + "&x=" +
x, ParameterType.RequestBody);
        IRestResponse response = client.Execute(request);

        //parse json and use the path url to login game
        JObject obj = JObject.Parse(response.Content);
        String path = (String) obj["path"];
        System.Console.WriteLine(path);
    }
}
```

```
public static string AESEncryptToString(object rawData) {
    return AESEncryptToString(JsonConvert.SerializeObject(rawData));
}

public static string AESEncryptToString(string JsonString) {

    var aESEncryptBytes = AESEncrypt(JsonString);
    var URL_Safe_AESEncrypt_String = Convert.ToBase64String(aESEncryptBytes)
                                                .TrimEnd(padding).Replace('+', '-').Replace('/', '_');
    return URL_Safe_AESEncrypt_String;
}

private static byte[] AESEncrypt(string plainText) {

    using(SymmetricAlgorithm des = Rijndael.Create()){

        byte[] inputByteArray = Encoding.UTF8.GetBytes(plainText);

        des.Key = Encoding.UTF8.GetBytes(AES_KEY);
        des.IV = Encoding.UTF8.GetBytes(AES_IV);

        var encrypter = des.CreateEncryptor();

        byte[] cipherBytes = encrypter.TransformFinalBlock(inputByteArray, 0,
inputByteArray.Length);

        return cipherBytes;
    }
}
}
```

## ➤ Sample – Decrypt:

```
using System;
using System.Text;
using System.Security.Cryptography;
using System.IO;

namespace StringEncrypt
{
    public static class StringEncrypt
    {
        public static void Main(string[] args) {
            string x = ${x}; // change it
            string key = ${aes_key}; // change it
            string iv = ${aes_iv}; // change it

            string plainText = aesDecryptBase64(x, key, iv);
            Console.WriteLine(plainText);
        }

        public static string aesDecryptBase64(string sourceStr, string keyStr, string ivStr)
        {
            string decrypt = "";
            try
            {
                AesCryptoServiceProvider aes = new AesCryptoServiceProvider();
                aes.BlockSize = 128;
                aes.Mode = CipherMode.CBC;
                aes.Padding = PaddingMode.None;
                byte[] key = Encoding.UTF8.GetBytes(keyStr);
                byte[] iv = Encoding.UTF8.GetBytes(ivStr);
                aes.Key = key;
                aes.IV = iv;

                string source = sourceStr.Replace('-', '+').Replace('_', '/');
                int addPaddingCounts = (4 - (source.Length % 4)) % 4;
                for (int i = 0 ; i < addPaddingCounts ; i++) {
                    source += "=";
                }
                byte[] dataByteArray = Convert.FromBase64String(source);
                using (MemoryStream ms = new MemoryStream())
                {
                    using (CryptoStream cs = new CryptoStream(ms,
                        aes.CreateDecryptor(), CryptoStreamMode.Write))
                    {
                        cs.Write(dataByteArray, 0, dataByteArray.Length);
                        cs.FlushFinalBlock();
                    }
                }
            }
            catch { }
            return decrypt;
        }
    }
}
```



```
        decrypt = Encoding.UTF8.GetString(ms.ToArray());
    }
}
}
catch (Exception e)
{
    Console.WriteLine(e.Message);
}
return decrypt;
}
}
}
```

### 3.4.Go Lang

#### ➤ Sample

```
package main

import (
    "bytes"
    "crypto/aes"
    "crypto/cipher"
    "encoding/base64"
    "errors"
    "fmt"
    "time"
    "strings"
    "strconv"
    "net/http"
    "io/ioutil"
)

const aesKeyDefValue = ${AES KEY}; // change it
const aesIvDefValue = ${AES IV}; // change it

func main() {
    // Action 11:required parameter information
    dc := ${DC}
    data := "{\"action\":11, \"ts\":\""+strconv.Itoa(int(time.Now().UnixNano()) / 1000000))+"\", \"uid\":\""+${uid}+"\", \"lang\":\"ch\", \"gType\":\"0\", \"mType\":\"8001\", \"windowMode\":\"2\"}"
    fmt.Println("data : ", data)

    // encrypt data
    encbyte, _ := AesEncrypt([]byte(data), checkAesKey(aesKeyDefValue))
    encryptString := strings.Replace(base64.StdEncoding.EncodeToString(encbyte), "+", "-", -1)

    // check environment Settings file
    apiURL := ${API_URL}
    payload := strings.NewReader("dc=" + string(dc) + "&x=" + encryptString)

    // post
    req, _ := http.NewRequest("POST", apiURL, payload)
    req.Header.Add("content-type", "application/x-www-form-urlencoded")
    req.Header.Add("cache-control", "no-cache")
    res, _ := http.DefaultClient.Do(req)
    defer res.Body.Close()
    body, _ := ioutil.ReadAll(res.Body)

    //parse json and use the path url to login game
    fmt.Println(string(body))
}
```

```
func checkAesKey(strKey string) []byte {
    keyLen := len(strKey)
    arrKey := []byte(strKey)

    if keyLen >= 32 {
        return arrKey[:32]
    }

    if keyLen >= 24 {
        return arrKey[:24]
    }

    if keyLen >= 16 {
        return arrKey[:16]
    }

    tmp := make([]byte, 16)
    for i := 0; i < 16; i++ {
        if i < keyLen {
            tmp[i] = arrKey[i]
        } else {
            tmp[i] = '0'
        }
    }
    return tmp
}

func AesEncrypt(plaintext []byte, key []byte) ([]byte, error) {
    block, err := aes.NewCipher(key)

    if err != nil {
        fmt.Println("err=", err)
        return nil, errors.New("invalid decrypt key")
    }

    blockSize := block.BlockSize()
    plaintext = PKCS5Padding(plaintext, blockSize)
    iv := []byte(aesIvDefValue)
    blockMode := cipher.NewCBCEncrypter(block, iv)

    ciphertext := make([]byte, len(plaintext))
    blockMode.CryptBlocks(ciphertext, plaintext)

    return ciphertext, nil
}
```

```
func AesDecrypt(ciphertext []byte, key []byte) ([]byte, error) {  
    block, err := aes.NewCipher(key)  
    if err != nil {  
        return nil, errors.New("invalid decrypt key")  
    }  
  
    blockSize := block.BlockSize()  
  
    if len(ciphertext) < blockSize {  
        return nil, errors.New("ciphertext too short")  
    }  
  
    iv := []byte(aesIvDefValue)  
    if len(ciphertext)%blockSize != 0 {  
        return nil, errors.New("ciphertext is not a multiple of the block size")  
    }  
  
    blockModel := cipher.NewCBCDecrypter(block, iv)  
  
    plaintext := make([]byte, len(ciphertext))  
    blockModel.CryptBlocks(plaintext, ciphertext)  
    plaintext = PKCS5UnPadding(plaintext)  
  
    return plaintext, nil  
}  
  
func PKCS5Padding(src []byte, blockSize int) []byte {  
    padding := blockSize - len(src)%blockSize  
    padtext := bytes.Repeat([]byte{byte(padding)}, padding)  
    return append(src, padtext...)  
}  
  
func PKCS5UnPadding(src []byte) []byte {  
    length := len(src)  
    unpadding := int(src[length-1])  
    return src[:length - unpadding]  
}
```

## 3.5.JavaScript

### ➤ AES Encryption

```
//util/aes.js
CryptoJS = require("crypto-js")

function encrypt(aesIv, aesKey, decryptedString) {
  const iv = CryptoJS.enc.Utf8.parse(aesIv)
  const key = CryptoJS.enc.Utf8.parse(aesKey)
  const encrypted = CryptoJS.AES.encrypt(decryptedString.trim(), key,
    {iv, padding: CryptoJS.pad.ZeroPadding}).toString()
  return base64EncodeUrl(encrypted)
}

function base64EncodeUrl(str) {
  return str.replace(/\+/g, '-').replace(/\//g, '_').replace(/=+$/, '')
}

module.exports = {encrypt};
```

```
//main.js
const aesUtil = require('./util/aes')
const axios = require('axios')
const qs = require('querystring')
const url = '{API URL}'
const axiosInstance = axios.create({
  headers: {'Content-Type': 'application/x-www-form-urlencoded'}
});

const dc = '{Your Domain}'
const iv = '{Your IV}'
const key = '{Your KEY}'
const request = {
  "action": 11,
  "ts": new Date().getTime(),
  'uid': '{UID}',
  'lang': 'en',
  'gType': '0',
  'mType': '8001',
  'windowMode': '2'
}
console.log(request)

const x = aesUtil.encrypt(iv, key, JSON.stringify(request));
console.log('x:', x)

axiosInstance.post(url, qs.stringify({dc, x}))
  .then(({data}) => {
    console.log('data:', data)
  })
```

## 4. FTP Functions

### 4.1. Function Description

- About FTP information, please refer to “JDB API Environment Setting Data.pdf”.
- The file of game transactions created every 5 minutes.
- The data is based on the last modified time of the game record
- For more detail, please refer to “[Action 29: Search for Details of Transactions in Games](#)”.
- A house can see game transactions of all players.
- A agent only see game transactions of the downline players.
- Manager, Master Agent does not provide FTP function.

### 4.2. Folder Path Description

- The folder path for the house  
/manager/master/agent/gameType/date
- The folder path for the agent  
/gameType/date/file
- The rule of file name  
yyyyMMddHHmm\_yyyyMMddHHmm.zip(start time\_end time)
- The file type is zipped file.

### 4.3. When create a new agent

- FTP account: agent account + “@” + DC
- FTP password: the first 6 digits is house’s password + agent account  
Ex: The house’s password is 1234567890 and the DC is AB, and the new agent is testag.  
FTP account: testag@AB  
FTP password: 123456testag

## 5. Appendix

### 5.1. Error Codes

Code	Description
0000	Succeed.
9999	Failed.
9001	No authorized to access
9002	Domain is null or the length of domain less than 2.
9003	Failed to pass the domain validation.
9004	The encrypted data is null or the length of the encrypted data is equal to 0.
9005	Assertion(SAML) didn't pass the timestamp validation.
9006	Failed to extract the SAML parameters from the encrypted data.
9007	Unknow action.
9008	The same value as before.
9009	Time out.
9010	Read time out.
9011	Duplicate transactions.
9012	Please try again later.
9013	System is maintained.
9014	Multiple account login detected.
9015	Data does not exist.
9017	Work in process, please try again later.
8000	The parameter of input error, please check your parameter is correct or not.
8001	The parameter cannot be empty.
8002	The parameter must be an positive integer.
8003	The parameter cannot be negative.
8005	wrong sdate second format
8006	Time does not meet.
8007	The parameter only can use number.
8008	The parameter cannot be found.
8009	Time interval exceeds the allowable range.
8010	The length of parameter is too long.
7001	The specified parent ID cannot be found.
7002	Parent is suspended.
7003	Parent is locked.
7004	Parent is closed.

<b>7405</b>	You have been logged out!
<b>7501</b>	User ID cannot be found.
<b>7502</b>	User is suspended.
<b>7503</b>	User is locked.
<b>7504</b>	User is closed.
<b>7505</b>	User is not playing
<b>7601</b>	Invalid User ID. Please only use characters between a-z, 0-9
<b>7602</b>	Account already exist. Please choose other User ID
<b>7603</b>	Invalid username.
<b>7604</b>	The password must at least 6 characters, with 1 alphabet and 1 number.
<b>7605</b>	Invalid operation_code. Please only use number 2, 3, 4, 5.
<b>6001</b>	Your Cash Balance not enough to withdraw
<b>6002</b>	User balance is zero
<b>6003</b>	Withdraw negative amount
<b>6004</b>	Duplicate Transfer
<b>6005</b>	Repeat serial number.
<b>6006</b>	Your Cash Balance not enough.
<b>6901</b>	User is playing game, and not allow transfer balance.



## 5.2.Currency Code

Code	ISO 4217 Code	Currency	Remark
AE	AED	Dirham	
AU	AUD	Australian dollar	
BN	BND	Brunei	
BR	BRL	Brazilian real	
BT	BDT	Taka, Bangladesh	
CA	CAD	Canadian Dollars	
CH	CHF	Swiss Franc	
CL	CLP	Chilean peso	
EU	EUR	Euro	
GB	GBP	Pound sterling	
HK	HKD	Hong Kong dollar	
JP	JPY	Japanese yen	
KW	KRW	South Korean won	
LK	LKR	Sri Lankan Rupee	
MM	MMK	Myanma kyat	1:100
MM2	MMK	Myanma kyat	
MX	MXN	Mexican peso	
NO	NOK	Norwegian Krone	
NZ	NZD	New Zealand currency	
PO	PO	Points	
PP	PHP	Philippine Bissau	
RB	RMB	China Yuan	
RM	MYR	Malaysian ringgit	
RP	IDR	Indonesian Shield	1:1000
RS	INR	Indian Rupee	
SE	SEK	Swedish Krona	
SG	SGD	Singapore Dollar	
TB	THB	Thai baht	
US	USD	United States dollar	
VN	VND	Vietnamese Shield	1:1000
ZA	ZAR	South African currency	
ZW	ZWD	Zimbabwean currency	