



# JDB Seamless Wallet

## API 规格书

Version: 1.12  
Date: 2019-09-18

## 修订历程

日期	版本	修订项目	修订者
2017-05-31	1.0	i. 创建文件	Brenda Hou
2017-07-07	1.1	i. 取得 token 移除 x, 新增 path	Roger Wu
2018-02-27	1.2	i. 更新 action 11, action 47: A. 增加验证 gType + mType 是否存在, 验证失败则回传错误代码: 8000 B. 新增 isAPP, lobbyURL 参数 ii. 新增功能 A. 查询游戏列表 Action 49 B. 查询使用者是否在游戏中 Action 52 C. 查询游戏结果 Action 54	Brenda Hou
2018-03-26	1.3	i. 修正错误代码「9011」: 重复交易	Brenda Hou
2018-04-10	1.4	i. 取消下注: 增加错误代码说明 ii. 追加查询单一钱包交易结果	Brenda Hou
2018-07-02	1.5	i. 新增功能 A. 赶出用户 Action 17 B. 赶出所有下线用户 Action 58	Brenda Hou
2018-07-27	1.6	i. 更新 action 21, action 47: 新增 moreGame, mute 参数 ii. 更新 Action 4: A. 增加带入参数 uid B. 删除带入参数 gType iii. 更新 Action 8: A. 增加带入参数 reportDate, roomType B. 调整带入参数信息依游戏类型区分	Brenda Hou
2018-08-02	1.7	i. 更新 Action 21: 新增 jackpotFlag 参数	Brenda Hou
2018-08-23	1.8	i. 调整 Action 54: 增加指定是否显示用户账号	Brenda Hou
2018-09-03	1.9	i. 新增 Action 62: 查询榜单活动	Roger Wu
2018-11-06	1.10	i. 修正文件跑版问题 ii. 修正 Action 8: gameSeqNo 回传型态由 Integer 改为 Long iii. 用户账号(uid, playerId)长度限制从 String(19) 改为 String(25)	Brenda Hou Amuro Huang
2019-03-21	1.11	i. 加强文件说明 ii. 修正错误代码英文说明: 8007, 7601 iii. 增加货币代码至 27 种 (包含点数)	Brenda Hou

		iv. 范例程序增加 Go Lang 版本	
2019-09-18	1.12	i. 范例程序新增 C# 解密范例 ii. Action 11, 47 新增栏位 isShowDollarSign	Amuro Huang

## 目录

1. 介绍.....	3
1.1. 注意事项.....	3
1.2. 必要功能及说明.....	3
2. 需提供以下 API 给 JDB 游戏平台 .....	4
2.1. 描述.....	4
2.2. 呼叫方式.....	4
2.3. 功能项目 .....	5
2.3.1. Action 4：取消下注 .....	5
2.3.2. Action 6：取得余额 .....	7
2.3.3. Action 8：下注信息及游戏结果 .....	9
3. JDB Platform API .....	16
3.1. 使用流程及注意事项.....	16
3.1.1. 呼叫方式.....	16
3.1.2. 回传格式.....	16
3.1.3. AES 加密注意事项 .....	16
3.2. 功能项目 .....	17
3.2.1. Action 17：赶出玩家 .....	17
3.2.2. Action 21：取得 Token.....	19
3.2.3. Action 42：每日对帐报表 .....	21
3.2.4. Action 43：开关 Jackpot 贡献值 .....	23
3.2.5. Action 45：查询 Jackpot 资讯.....	25
3.2.6. Action 47：取得试玩 Token.....	27
3.2.7. Action 49：查询游戏列表 .....	29
3.2.8. Action 52：查询玩家是否在游戏中 .....	31
3.2.9. Action 54：查询游戏结果 .....	33
3.2.10. Action 58：赶出所有下线用户 .....	35
3.2.11. Action 59：查询单一钱包交易结果 .....	37
3.2.12. Action 62：查询活动榜单 .....	39
4. 范例程序.....	41
4.1. Java .....	41
4.2. PHP .....	45
4.3. C#.....	47
4.4. Go Lang.....	51
5. 附录.....	54
5.1. 错误代码.....	54

5.2. 货币代码.....	56
----------------	----

# 1. 介绍

## 1.1. 注意事项

- API 网址、加密金鑰与游戏列表等信息，请洽本公司业务索取。
- 使用此系统时，需提供贵司联机的服务器 IP
- 回传格式仅提供 JSON 格式

## 1.2. 必要功能及说明

- 需提供 API 給 JDB 遊戲平台
  - 取得餘額
    - ◆ 请参考「[Action 6：取得余额](#)」
  - 下注信息及遊戲結果
    - ◆ 请参考「[Action 8：下注信息及游戏结果](#)」
  - 取消下注
    - ◆ 请参考「[Action 4：取消下注](#)」
- JDB 提供的 API
  - 進行遊戲
    - ◆ 一般玩家
      - 利用段落「[Action 21：取得 Token](#)」方式取得 path
      - 直接导向此 path 即可登入系统并进行游戏
    - ◆ 试玩玩家
      - 利用段落「[Action 47：取得试玩 Token](#)」方式取得 path
      - 直接导向此 path 即可登入系统并进行游戏
  - 查詢單一錢包交易結果
    - ◆ 请参考「[Action 59：查询单一钱包交易结果](#)」

## 2. 需提供以下 API 给 JDB 游戏平台

### 2.1. 描述

此 API 调用次数不限，可用于下注及派彩在营运系统情况下，请确保这个超时设定于 2 秒内，3 秒后为超时，即使最新的系统回传，也能使在超时前接收得到。

有关于超时的状况，可以重新调用（JDB 系统会重复调用三次）。

### 2.2. 呼叫方式

由网站营运商提供 { **HttpUrl** }，我们可使用这个 API 把信息返回。

➤ API URL

{ **HttpUrl** }

➤ HTTP Method: POST

参数	值
<b>x</b>	<ol style="list-style-type: none"><li>1. 参阅章节「<a href="#">功能项目</a>」决定要呼叫的功能</li><li>2. 依功能所需的参数转为 JSON String</li><li>3. 将此 JSON String 利用 AES 128 bit 进行解密</li><li>4. 解密步骤，可参考「<a href="#">范例程序</a>」</li></ol>

## 2.3.功能项目

### 2.3.1. Action 4：取消下注

➤ 功能说明

当下注请求(Action 8)失败时呼叫该 API。

➤ 参数

参数	格式	说明
<b>action</b>	Integer	4
<b>ts</b>	Long	当前系统时间
<b>transferId</b>	Long	交易序号
<b>uid</b>	String	玩家账号，只限 a-z 与 0-9。

Example:

```
{
  "action":4,
  "ts":1447452951820,
  "transferId":"123456789",
  "uid":"testpl01"
}
```



## ➤ 返回结果

参数	格式	说明
<b>status</b>	String	成功: 0000 错误: 9015:交易序号不存在(参照附录「 <a href="#">错误代码</a> 」) 如未符合需求, 则设定为 <b>9999</b> , 并在 <b>err_text</b> 说明错误原因
<b>balance</b>	Double	余额
<b>err_text</b>	String	错误讯息

## ➤ 操作范例

## Http Request

```
http://{ HttpUrl }?x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-UWsf6PHz9Ky-
wM3mIC9E14kw1_S1X3lpraVaCLnp-
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU
gTK5yzbwKEddANjAAbxF0s=
```

## Http Response (success)

```
{
  "status": "0000",
  "balance": 9990
}
```

## Http Response (error)

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

## 2.3.2. Action 6：取得余额

- 功能说明  
回传玩家剩余金额。

- 参数

参数	格式	说明
<b>action</b>	Integer	6
<b>ts</b>	Long	当前系统时间
<b>uid</b>	String	玩家账号，只限 a-z 与 0-9。
<b>currency</b>	String	货币别

Example:

```
{  
  "action":6  
  "ts":1447452951820,  
  "uid":"testpl01",  
  "currency":"RB"  
}
```

## ➤ 返回结果

参数	格式	说明
<b>status</b>	String	成功: 0000 错误: 7501-玩家不存在 ※参照附录「 <a href="#">错误代码</a> 」, 如未符合需求, 则设定为 9999, 并在 <b>err_text</b> 说明错误原因
<b>balance</b>	Double	余额
<b>err_text</b>	String	错误讯息

## ➤ 操作范例

## Http Request

```
http://{ HttpUrl }?x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-Uwsfd6PHz9Ky-
wM3mIC9E14kw1_S1X3lpraVaCLnp-
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU
gTK5yzbWKEddANjAAbxF0s=
```

## Http Response (success)

```
{
  "status": "0000",
  "balance": 10000
}
```

## Http Response (error) ※玩家余额不足

```
{
  "status": "6006",
  "err_text": "Your Cash Balance not enough."
}
```

## 2.3.3. Action 8：下注信息及游戏结果

### ➤ 功能说明

- 当玩家下注时呼叫该 API，通知该局游戏结果。
- 确认玩家余额是否足够，并请进行扣款后，回传玩家剩余金额。
- 常見錯誤代碼說明
  - ✓ 9011：交易序号重复
  - ✓ 7501：玩家不存在
  - ✓ 6002：玩家余额为零
  - ✓ 6006：玩家余额不足
  - ✓ 其他错误代码请参照附录「[错误代码](#)」

### ➤ 参数

参数	格式	说明
action	Integer	8
ts	Long	当前系统时间
transferId	Long	交易序号
gameSeqNo	Long	游戏序号.
uid	String	玩家账号，只限 a-z 与 0-9。
gType	Integer	游戏型态(请参考「JDB Games.pdf」)
mType	Integer	机台类型(请参考「JDB Games.pdf」)
reportDate	String	报表日期 (dd-MM-yyyy)
gameDate	String	游戏日期 (dd-MM-yyyy HH:mm:ss)
currency	String	货币别
bet	Double	押注金额
win	Double	游戏赢分
netWin	Double	总输赢
denom	Double	投注面值
ipAddress	String	玩家登入 IP
clientType	String	玩家从网页或行动装置登入
systemTakeWin	Integer	系统结算标记 0: 否 1: 是
lastModifyTime	String	最后修改时间 (dd-MM-yyyy HH:mm:ss)
老虎机下注记录: gType=0		
jackpotWin	Double	赢得彩金金额
jackpotContribute	Double	彩金贡献值
hasFreeware	Integer	免费游戏

		0: 否 1: 是
<b>hasGamble</b>	Integer	博取游戏 0: 否 1: 是
<b>鱼机下注记录: gType=7</b>		
<b>roomType</b>	Integer	游戏区域 0: 欢乐 1: 富豪 2: 依机台类型区分 mType=7003: 财神 mType=7004: 五龙
<b>街机下注记录: gType=9</b>		
<b>hasGamble</b>	Integer	博取游戏 0: 否 1: 是
<b>hasBonusGame</b>	Integer	奖金游戏 0: 否 1: 是
<b>电子彩票下注记录: gType=12</b>		
<b>hasBonusGame</b>	Integer	奖金游戏 0: 否 1: 是

## Example – 老虎机:

```
{  
  "action":8,  
  "ts":1447452951820,  
  "transferId":1001,  
  "gameSeqNo":5250145705663,  
  "uid":"test01",  
  "gType ":0,  
  "mType":8001,  
  "reportDate":"13-04-2017",  
  "gameDate":"11-07-2018 11:37:39",  
  "currency":"RB",  
  "bet":-5.25,  
  "win":2,  
  "netWin":-3.25,  
  "denom":0.01,  
  "ipAddress":"10.20.6.86",  
  "clientType":"Web",  
  "systemTakeWin":0,  
  "lastModifyTime":"11-07-2018 11:37:45",  
  "jackpotWin":0,  
  "jackpotContribute":-0.00125,  
  "hasFreegame":0,  
  "hasGamble":0  
}
```

## Example – 鱼机:

```
{  
  "action":8,  
  "ts":1447452951820,  
  "transferId":1001,  
  "gameSeqNo":5250145705663,  
  "uid":"test01",  
  "gType ":7,  
  "mType":7001,  
  "reportDate":"13-04-2017",  
  "gameDate":"11-07-2018 11:37:39",  
  "currency":"RB",  
  "bet":-5.25,  
  "win":2,  
  "netWin":-3.25,  
  "denom":0.01,  
  "ipAddress":"10.20.6.86",  
  "clientType":"Web",  
  "systemTakeWin":0,  
  "lastModifyTime":"11-07-2018 11:37:45",  
  "roomType":0  
}
```

## Example – 街机:

```
{  
  "action":8,  
  "ts":1447452951820,  
  "transferId":1001,  
  "gameSeqNo":5250145705663,  
  "uid":"test01",  
  "gType":9,  
  "mType":9001,  
  "reportDate":"13-04-2017",  
  "gameDate":"11-07-2018 11:37:39",  
  "currency":"RB",  
  "bet":-5.25,  
  "win":2,  
  "netWin":-3.25,  
  "denom":0.01,  
  "ipAddress":"10.20.6.86",  
  "clientType":"Web",  
  "systemTakeWin":0,  
  "lastModifyTime":"11-07-2018 11:37:45",  
  "hasGamble":0,  
  "hasBonusGame":0  
}
```



## Example – 电子彩票:

```
{
  "action":8,
  "ts":1447452951820,
  "transferId":1001,
  "gameSeqNo":5250145705663,
  "uid":"test01",
  "gType ":12,
  "mType":12001,
  "reportDate":"13-04-2017",
  "gameDate":"11-07-2018 11:37:39",
  "currency":"RB",
  "bet":-5.25,
  "win":2,
  "netWin":-3.25,
  "denom":0.01,
  "ipAddress":"10.20.6.86",
  "clientType":"Web",
  "systemTakeWin":0,
  "lastModifyTime":"11-07-2018 11:37:45",
  "hasBonusGame":0
}
```

## ➤ 返回结果

参数	格式	说明
<b>status</b>	String	成功: 0000 错误: 9011-交易序号重复 7501-玩家不存在 6002-玩家余额为零 6006-玩家余额不足 ※参照附录「错误代码」, 如未符合需求, 则设定为 9999, 并在 err_text 说明错误原因
<b>balance</b>	Double	余额
<b>err_text</b>	String	错误讯息

## ➤ 操作范例

## Http Request

```
http://{ HttpUrl }?x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-UWsf6PHz9Ky-
wM3mIC9E14kw1_S1X3lpraVaCLnp-
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU
gTK5yzbWKEddANjAAbxF0s=
```

## Http Response (success)

```
{
  "status": "0000",
  "balance": 10000
}
```

## Http Response (error) ※玩家余额不足

```
{
  "status": "6006",
  "err_text": "Your Cash Balance not enough."
}
```

## 3. JDB Platform API

### 3.1. 使用流程及注意事项

#### 3.1.1. 呼叫方式

- API URL  
http:// {API\_Address}/apiRequest.do
- HTTP Method: POST

参数	值
dc	客户域名
x	<ol style="list-style-type: none"><li>1. 参阅章节「<a href="#">功能项目</a>」决定要呼叫的功能</li><li>2. 依功能所需的参数转为 JSON String</li><li>3. 将此 JSON String 利用 AES 128 bit 进行加密</li><li>4. 加密步骤，可参考「<a href="#">范例程序</a>」</li></ol>

#### 3.1.2. 回传格式

- 仅提供 JSON 格式
- **HTTP 压缩**: 用户可选择是否要压缩回传的讯息。  
如要压缩，需在 Http Request 的 header 中加入 “Accept-Encoding”: gzip 设定

#### 3.1.3. AES 加密注意事项

- 加密数据**有效期为 30 秒**，如果在 30 秒内未发送，则数据将会过期。  
**逾期时会收到 9005 的错误代码，请校正系统时间。**
- 由于 AES 加密过程，是针对每个固定大小的区块进行转换，为了让加密能够顺利进行，必须将数据的 size 补齐到能够被 **16 bytes** 整除的大小，若**不足时请使用空字符串向右补齐**。

## 3.2. 功能项目

### 3.2.1. Action 17: 赶出玩家

➤ 功能说明

- 如果玩家在线，玩家将会收到一封被赶出的提示讯息。
- 如果玩家正在进行游戏，游戏将会被中止并直接结算。

➤ 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	17
<b>ts</b>	Long	Y	当前系统时间
<b>parent</b>	String(19)	Y	代理账号
<b>uid</b>	String(25)	Y	玩家账号

Example:

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

## ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>err_text</b>	String(255)	错误讯息

## ➤ 操作范例

## Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-  
UWsfd6PHz9Ky-wM3mIC9E14kw1_SlX3lpraVaCLnp-  
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU  
gTK5yzbWKEddANjAAbxF0s=
```

## Http Response (success)

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

## Http Response (error)

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

### 3.2.2. Action 21：取得 Token

➤ 功能说明

此功能提供用户在登入系统时须使用的 Token。

➤ 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	21
<b>ts</b>	Long	Y	当前系统时间
<b>parent</b>	String(19)	Y	代理账号
<b>uid</b>	String(25)	Y	玩家账号
<b>balance</b>	Double	Y	余额
<b>lang</b>	String(2)	N	语系 en: 英文（默认值） ch: 简体中文
<b>gType</b>	String(1)	N	游戏型态
<b>mType</b>	String(3)	N	机台类型
<b>windowMode</b>	String(1)	N	1: 使用 JDB 游戏大厅（默认值） ※若未带入 gType 及 mType，则直接到游戏大厅 ※若带入 gType 及 mType 时，直接进入游戏。  2: 不使用 JDB 游戏大厅 ※gType 及 mType 为必填字段。
<b>isAPP</b>	Boolean	N	是否为手机 APP 进入游戏 true: 手机 APP false: 手机网页、计算机网页（默认值）
<b>lobbyURL</b>	String(1000)	N	游戏大厅网址 当 windowMode 为 2 时，此参数才会有作用
<b>moreGame</b>	Integer	N	0: 不显示更多游戏 1: 显示更多游戏（默认值）
<b>mute</b>	Integer	N	默认音效开关 0: 开启音效（默认值） 1: 静音
<b>jackpotFlag</b>	Integer	N	设定彩金开关 0: 打开（默认值） 1: 关闭
<b>isShowDollarSign</b>	Boolean	N	是否显示币别符号

		<b>true:</b> 显示币别符号 <b>false:</b> 不显示币别符号
--	--	--

Example:

```
{
  "action":21,
  "ts":1447490495704,
  "parent":"testag",
  "uid":"testpl01",
  "balance":10000,
  "gType":0,
  "mType":8001
}
```

### ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>err_text</b>	String(255)	错误讯息
<b>path</b>	String(255)	开启游戏连结

### ➤ 操作范例

#### Http Request

http://{API\_Address}/apiRequest.do?dc=jb1&x=y1wayxScczy6ABfElfaVwb5CdfHxBN\_JGtxyBSvpohLQakP8M2QSetJNojJBzvrVChhq8YWWgBTdGeXFT4vSNQQ66epf4NqYeQ-6bJ15pvHctQjQevulv7GHqOifuju\_gY0H54FfVdQc9D5tKh1AZ-9KDh9dwcGSszZxU7JOOoHdYITICO5LhvGmHxPBwZ

#### Http Response (success)

```
{
  "status":"0000",
  "path":"http://xxx.xxx.xxx?x=Q0Y2NUZDNEFBODEzNUM5NURGRjc5QTUzRDk0R
TlDRDUzNEQ0MjhEOEMzMURDODBCNzdCMEE2OTE3MzcZrKE5RDNGQ0QzQjdFNTlB
MjE4Njg5NDM4RDREMEI1NkYzQjkxOEU5QTU0QTM5QTY1MEUYnjYzMDg1RkFBQTM
1OUU3RTI2RTBGQUEzMUI0MkQ4NEFBRjVDNkJFRkQzMzU1NzA5OUNDMkYxMzMMyOT
hBNjhBRURCNzExREMzRjhDNzA1N0VFNEIzMzhCNTFEQ0FDODVEODQ3NEFDNTE1M
EE1NTMzODQ="
}
```

#### Http Response (error)

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

### 3.2.3. Action 42：每日对帐报表

#### ➤ 功能说明

- 查询单一代理在特定日期内的游戏详细交易信息
- 查询日期为 01-01-2016，则取得数据范围为 01-01-2016 12:00:00 至 01-02-2016 12:00:00。
- 每次查询最新信息为 1 天前的资料。
  - ◆ 例 1：系统时间为 05-01-2016 18:23:00，则查询日期需小于等于 04-01-2016
  - ◆ 例 2：系统时间为 05-01-2016 09:23:02，则查询日期需小于等于 03-01-2016（因 04-01-2016 的报表日期还在进行中）
  - ◆ 例 3. 系统时间为 05-01-2016 12:11:22，则查询日期需小于等于 03-01-2016（为确保数据完整性，延后 1 小时取值）
- 每次查询只提供范围为三个月内的资料
- 启用棋牌游戏时，所有游戏态型皆增加回传「tax」讯息。

#### ➤ 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	42
<b>ts</b>	Long	Y	当前系统时间
<b>parent</b>	String(19)	Y	代理账号
<b>gType</b>	String(1)	Y	游戏型态
<b>date</b>	String(10)	Y	查询日期(dd-MM-yyyy)

Example:

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



## ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」 ※查无资料: 0000
<b>data</b>	JSON Object	
<b>uid</b>	String(25)	玩家账号
<b>bet</b>	Double	押注金额
<b>win</b>	Double	游戏赢分
<b>netWin</b>	Double	总输赢
<b>jackpot</b>	Double	赢得彩金金额
<b>jackpotContribute</b>	Double	彩金贡献值
<b>count</b>	Integer	笔数
<b>validBet</b>	Double	有效押注金额
<b>err_text</b>	String(255)	错误讯息

## ➤ 操作范例

## 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": "testpl01", "bet": -3760.8, "win": 2590.6, "netWin": -1170.2, "jackpot": 0, "jackpotContribute": -18.804, "count": 238, "validBet": -3760.8},
    {"uid": "testpl02", "bet": -2796.2, "win": 1789.65, "netWin": -1006.55, "jackpot": 0, "jackpotContribute": -13.981, "count": 274, "validBet": -2796.2},
    {"uid": "testpl03", "bet": -0.5, "win": 0.05, "netWin": -0.45, "jackpot": 0, "jackpotContribute": -0.0025, "count": 1, "validBet": -0.5}
  ]
}
```

## Http Response (error)

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

### 3.2.4. Action 43：开关 Jackpot 贡献值

- 功能说明  
针对部份玩家关闭 Jackpot 的功能

- 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	43
<b>ts</b>	Long	Y	当前系统时间
<b>parent</b>	String(19)	Y	代理账号
<b>uid</b>	String(25)	Y	玩家账号
<b>jackpotFlag</b>	Integer	Y	0: 打开 1: 关闭

Example:

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

## ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>err_text</b>	String(255)	错误讯息

## ➤ 操作范例

## Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-  
UWsfd6PHz9Ky-wM3mIC9E14kw1_SlX3lpraVaCLnp-  
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU  
gTK5yzbwKEddANjAAbxF0s=
```

## Http Response (success)

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

## Http Response (error)

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

### 3.2.5. Action 45：查詢 Jackpot 資訊

#### ➤ 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	45
<b>ts</b>	Long	Y	当前系统时间
<b>parent</b>	String(19)	Y	代理账号

Example:

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

#### ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>grand</b>	JSON Object	
<b>val</b>	Double	巨奖目前金额
<b>max</b>	Double	巨奖最大金额
<b>min</b>	Double	巨奖最小金额
<b>major</b>	JSON Object	
<b>val</b>	Double	大奖目前金额
<b>max</b>	Double	大奖最大金额
<b>min</b>	Double	大奖最小金额
<b>minor</b>	JSON Object	
<b>val</b>	Double	小奖目前金额
<b>max</b>	Double	小奖最大金额
<b>min</b>	Double	小奖最小金额
<b>err_text</b>	String(255)	错误讯息

➤ 操作范例

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-  
UWsfd6PHz9Ky-wM3mIC9E14kw1_SlX3lpraVaCLnp-  
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU  
gTK5yzbwKEddANjAAbxF0s=
```

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."  
}
```

### 3.2.6. Action 47：取得试玩 Token

➤ 功能说明

此功能提供用户试玩登入系统时须使用的 Token。

➤ 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	47
<b>ts</b>	Long	Y	当前系统时间
<b>lang</b>	String(2)	N	语系 en: 英文（默认值） ch: 简体中文
<b>gType</b>	String(2)	N	游戏型态
<b>mType</b>	String(5)	N	机台类型 ※若需直接进入游戏，需同时带入 mType 及 gType
<b>windowMode</b>	String(1)	N	1: 包含游戏大厅（默认值） ※若未带入 gType 及 mType，则直接到游戏大厅 ※若带入 gType 及 mType 时，直接进入游戏。  2: 不包含游戏大厅，隐藏游戏中的关闭按钮 ※gType 及 mType 为必填字段。
<b>isAPP</b>	Boolean	N	是否为手机 APP 进入游戏 true: 手机 APP false: 手机网页、计算机网页（默认值）
<b>lobbyURL</b>	String(1000)	N	游戏大厅网址 当 windowMode 为 2 时，此参数才会有作用
<b>moreGame</b>	Integer	N	0: 不显示更多游戏 1: 显示更多游戏（默认值）
<b>mute</b>	Integer	N	默认音效开关 0: 开启音效（默认值） 1: 静音
<b>isShowDollarSign</b>	Boolean	N	是否显示币别符号 true: 显示币别符号 false: 不显示币别符号

## Example:

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

## ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>err_text</b>	String(255)	错误讯息
<b>path</b>	String(255)	登入 URL

## ➤ 操作范例

## Http Request

http://{API\_Address}/apiRequest.do?dc=jb1&x=y1wayxScczy6ABfElfaVwb5CdfHxBN\_JGtxyBSvpohLQakP8M2QSetJNojJBzvnrvChhq8YWWgBTdGeXFTH4vSNQQ66epf4NqYeQ-6bJ15pvHctQjQevulv7GHqOifuju\_gY0H54FfVdQc9D5tKh1AZ-9KDh9dwcGSszZxU7JOOoHdYITICO5LhvGmHxPBwZ

## Http Response (success)

```
{
  "status":"0000",
  "path":"http://xxx.xxx.xxx/?x=Q0Y2NUZDNEFBODEzNUM5NURGRjc5QTUzRDk0RTlDRDUzNEQ0MjhEOEMzMURDODBCNzdCMEE2OTE3MzczRkE5RDNGQ0QzQjdFNTlBMjE4Njg5NDM4RDREMEI1NkYzQjkxOEU5QTU0QTM5QTY1MEUyNjYzMDg1RkFBQTm1OUU3RTI2RTBGQUEzMUI0MkQ4NEFBRjVDNkJFRkQzMzU1NzA5OUNDMkYxMzMzY0ThBNjhBRURCNzExREMzRjhDNzA1N0VFNEIzMzhCNTFEQ0FDODVEODQ3NEFDNTE1MEE1NTMzODQ="
}
```

## Http Response (error)

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

### 3.2.7. Action 49：查询游戏列表

- 功能说明
  - 取得游戏列表

- 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	49
<b>ts</b>	Long	Y	当前系统时间
<b>parent</b>	String(19)	Y	代理账号
<b>lang</b>	String(2)	N	en: 英文（默认值） ch: 简体中文

Example:

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

- 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>data</b>	JSON array	
<b>gType</b>	Integer	游戏型态
<b>list</b>	JSON array	
<b>mType</b>	Integer	机台类型
<b>isNew</b>	Boolean	是否为新游戏
<b>image</b>	String	游戏图片网址
<b>name</b>	String	游戏名称
<b>err_text</b>	String(255)	错误讯息



## ➤ 操作范例

**Http Request**

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-  
UWsfd6PHz9Ky-wM3mIC9E14kw1_S1X3lpraVaCLnp-  
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpgY0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU  
gTK5yzbwKEddANjAAbxF0s=
```

**Http Response (success)**

```
{  
  "status": "0000",  
  "data": [  
    {  
      "gType": 0,  
      "list": [  
        {  
          "mType": 8001,  
          "isNew": false,  
          "image": "https://xxx.xxx.xxx/images/xxx.jpg",  
          "name": "Game 1"  
        }  
      ]  
    },  
    {  
      "gType": 0,  
      "list": [  
        {  
          "mType": 8002,  
          "isNew": true,  
          "image": "https://xxx.xxx.xxx/images/xxx.jpg",  
          "name": "Game 2"  
        }  
      ]  
    }  
  ]  
}
```

**Http Response (error)**

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

### 3.2.8. Action 52：查询玩家是否在游戏中

#### ➤ 功能说明

- 查询使用者是否在游戏中，如在游戏中则回传相关信息

#### ➤ 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	52
<b>ts</b>	Long	Y	当前系统时间
<b>parent</b>	String(19)	Y	代理账号
<b>uid</b>	String(25)	Y	玩家账号

Example:

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

#### ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>data</b>	JSON array	
<b>gType</b>	Integer	游戏型态
<b>mType</b>	Integer	机台类型
<b>loginFrom</b>	String(20)	玩家从网页或行动装置登入
<b>ipAddr</b>	String(50)	玩家登入 IP
<b>loginTime</b>	String(19)	登入游戏时间 (dd-MM-yyyy HH:mm:ss)
<b>balance</b>	Number	账户余额
<b>err_text</b>	String(255)	错误讯息

➤ 操作范例

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-  
UWsfd6PHz9Ky-wM3mIC9E14kw1_SlX3lpraVaCLnp-  
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU  
gTK5yzbwKEddANjAAbxF0s=
```

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"  
}
```

### 3.2.9. Action 54：查询游戏结果

➤ 功能说明

- 查询玩家游戏结果
- 未带入 gType 与 seqNo 会进入报表主页
- 有带入 gType 与 seqNo 会直接进入该次游戏的详细信息页面

➤ 参数

参数	格式	强制	说明
action	Integer	Y	54
ts	Long	Y	当前系统时间
parent	String(19)	Y	代理账号
uid	String(25)	Y	玩家账号
lang	String(5)	N	en: 英文（默认值） ch: 简体中文
gType	Integer	N	游戏型态 若有带入 seqNo, 则 gType 为必填
seqNo	Long	N	注单号码 若有带入 gType, 则 seqNo 为必填
showUid	Integer	N	指定是否显示玩家账号 0: 不显示 1: 显示（默认值）

Example:

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

## ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>data</b>	JSON array	
<b>path</b>	String	游戏结果 URL
<b>err_text</b>	String(255)	错误讯息

## ➤ 操作范例

## Http Request

http://{API\_Address}/apiRequest.do?dc=jb1&x=y1wayxScczy6ABfElfaVwb5CdfHxBN\_JGtxyBS  
vpohLQakP8M2QSetJNojJBzvrVChhq8YWWgBTdGeXFTH4vSNQQ66epf4NqYeQ-  
6bJ15pvHctQjQevulv7GHqOifuju\_gY0H54FfVdQc9D5tKh1AZ-  
9KdH9dwcGSszZxU7JOOoHdYITICO5LhvGmHxPBwZ

## Http Response (success)

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

## Http Response (error)

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

### 3.2.10. Action 58: 赶出所有下线用户

#### ➤ 功能说明

- 此 action 会对 parent 的所有下线进行动作
- 如果玩家在线，玩家将会收到一封被赶出的提示讯息。
- 如果玩家正在进行游戏，游戏将会被中止并直接结算。
- 如果动作正在执行中，则再次呼叫会回传错误代码 9017

#### ※範例說明

若 parent 为 house 账号，则此 house 账号所有下线皆会被赶出系统（包含经理，代理，总代理，会员）

若 parent 为 agent 账号，则此 agent 账号下所有会员皆会被赶出系统。

#### ➤ 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	58
<b>ts</b>	Long	Y	当前系统时间
<b>parent</b>	String(19)	Y	上线账号

#### Example:

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

## ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>err_text</b>	String(255)	错误讯息

## ➤ 操作范例

## Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-  
UWsfd6PHz9Ky-wM3mIC9E14kw1_SlX3lpraVaCLnp-  
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU  
gTK5yzbwKEddANjAAbxF0s=
```

## Http Response (success)

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

## Http Response (error)

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

### 3.2.11. Action 59：查询单一钱包交易结果

#### ➤ 功能说明

- 查询在特定时间内的单一钱包事务历史记录
- 开始时间与结束时间中，ss(秒数)的值必须为 00
- 每次查询时间范围最多为 15 分钟，最新信息为 3 分钟前。例如，在 15:00:00 时的查询，最晚只能查到 14:56:00~14:57:00 的信息。
- 建议每次查询范围不超过 1 分钟，以获得较佳效果。
- 提供 2 小时内的交易结果。

#### ➤ 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	59
<b>ts</b>	Long	Y	当前系统时间
<b>starttime</b>	String(19)	Y	查询区段起始时间 (dd-MM-yyyy HH:mm:ss)
<b>endtime</b>	String(19)	Y	查询区段结束时间 (dd-MM-yyyy HH:mm:ss)

Example:

```
{
  "action":59,
  "ts":1447452951820,
  "starttime":"08-12-2016 17:45:00",
  "endtime":"08-12-2016 17:46:00"
}
```



## ➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」
<b>data</b>	JSON Object	
<b>transferId</b>	Long	交易序号
<b>uid</b>	String	玩家账号
<b>gameDate</b>	String	游戏日期 (dd-MM-yyyy HH:mm:ss)
<b>createTime</b>	String	建立时间 (dd-MM-yyyy HH:mm:ss)
<b>flag</b>	Integer	交易结果 0: 失败 1: 成功
<b>err_text</b>	String(255)	错误讯息

## ➤ 操作范例

## Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-
UWsfD6PHz9Ky-wM3mIC9E14kw1_SlX3lpraVaCLnp-
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU
gTK5yzbWKEddANjAAbxF0s=
```

## Http Response (success)

```
{
  "status": "0000",
  "data": [
    {
      "transferId": 3003,
      "uid": "testplayer01",
      "gameDate": "13-04-2017 11:37:39",
      "createTime": "13-04-2017 11:37:49",
      "flag": 1
    },
    {
      "transferId": 3004,
      "uid": "testplayer01",
      "gameDate": "13-04-2017 11:37:39",
      "createTime": "13-04-2017 11:37:55",
      "flag": 0
    }
  ]
}
```

## Http Response (error)

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

### 3.2.12. Action 62: 查询活动榜单

➤ 功能说明

- 依活动日期区间查询榜单。

➤ 参数

参数	格式	强制	说明
<b>action</b>	Integer	Y	62
<b>ts</b>	Long	Y	当前系统时间
<b>startDate</b>	String(10)	Y	查询区段起始日期 (dd-MM-yyyy)
<b>endDate</b>	String(10)	Y	查询区段结束日期 (dd-MM-yyyy)

Example:

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

➤ 返回结果

参数	格式	说明
<b>status</b>	String(4)	成功: 0000 错误: 参照附录「 <a href="#">错误代码</a> 」 ※查无资料: 9015
<b>data</b>	JSON array	
<b>activityDate</b>	String(10)	活动日期 dd-mm-yyyy
<b>activityNo</b>	String(20)	活动代码
<b>activityName</b>	String(100)	活动名称
<b>awardData</b>	JSON array	
<b>awardId</b>	String(3)	奖项代码
<b>awardName</b>	String(50)	奖项名称
<b>rankData</b>	JSON array	
<b>rank</b>	Int	奖项名次
<b>uid</b>	String(25)	玩家账号
<b>amount</b>	Double	奖金
<b>parent</b>	String(19)	代理账号
<b>err_text</b>	String(255)	错误讯息

➤ 操作范例

Http Request

```
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-
UWsfd6PHz9Ky-wM3mIC9E14kw1_SlX3lpraVaCLnp-
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj70vpRw_iU4554RuSsvHpI9jaj4XU
gTK5yzbWKEddANjAAbxF0s=
```

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, "uid": "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."
}
```

## 4. 范例程序

### 4.1.Java

➤ 说明：

- 以 java 程序、登入系统交易信息为例
- 此范例包含三大部分：
  - ◆ AES 加密：给定 request 需要的数据与 iv、key，用 AES 加密。
  - ◆ 加密并发送：给定 request 需要的数据，启动上述的加密程序，并将请求发送至 server。
  - ◆ 测试程序：设定 request 需要的数据，启动上述的加密并发送程序，接收结果。

➤ 范例

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

public class MCrypt {
    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);
    }
}
```

## ➤ 解密

```
public static String decrypt(String data, String key, String iv) throws Exception {  
    Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding");  
    cipher.init(Cipher.DECRYPT_MODE, new SecretKeySpec(key.getBytes(), "AES"),  
        new IvParameterSpec(iv.getBytes()));  
    String plainText = new String(cipher.doFinal(Base64.decodeBase64(encData)));  
  
    return plainText;  
}
```

## ➤ 加密并发送

```
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;  
    }  
}
```

➤ 测试程序 - Action 21 取得 Token

```
@Test
public void testPerformAction21() throws Exception {
    // prepare data
    JsonObject v = generateNewVector();
    v.addProperty("action", 21);
    v.addProperty("ts", ${system time});
    v.addProperty("parent", ${parent id});
    v.addProperty("uid", ${player id});
    v.addProperty("balance", ${balance});
    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"));
}
```

## 4.2.PHP

- 说明：
  - 以 php7.0 程序、登入系统交易信息为例
- AES 加密、解密 (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, $iv);
        return base64_encode($encrypted);
    }

    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;
    }
}
?>
```



## ➤ 登入系统

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

    // Action 21:required parameter information
    $mc = new MCrypt();
    $now = round(microtime(true)*1000);
    $dc = ${DC};
    $uid = 'playerId';
    $jsonString = '{"action":21, "ts":'.$now.', "parent":"agent", "uid": "'.$uid."',
    "balance":100, "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>";

?>
```

### 4.3.C#

➤ 说明:

- 以 c# 程序进行 AES 加密及登入系统交易信息为例

➤ AES 加密及登入系统

```
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 21:required parameter information
        String dc = ${DC};
        String uid = ${uid};
        String jsonString = "{\"action\":\"21\", \"ts\":\"" +
DateTimeOffset.UtcNow.ToUnixTimeMilliseconds() + ", \"parent\": " + ${parent} + ", \"uid\": " +
uid + ", \"balance\":100, \"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;
    }
}
}
```

## ➤ AES 解密

```
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;
}
}
}
```

## 4.4.Go Lang

➤ 说明:

- 以 go lang 程序进行 AES 加密及登入系统交易信息为例

➤ 范例

```
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 := ${DC}
    data := "{\"action\":21, \"ts\":\""+strconv.Itoa(int(time.Now().UnixNano() / 1000000))+",
    \"parent\": \"\"+${agentId}+"\", \"uid\": \"\"+${uid}+"\", \"windowMode\":100, \"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)
    fmt.Println("afterEncode : ", encryptString)

    // 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]  
}
```



## 5. 附录

### 5.1. 错误代码

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. 货币代码

代码	ISO 4217 Code	货币
AU	AUD	澳币
BN	BND	汶莱币
BT	BDT	孟加拉国塔卡
CA	CAD	加币
CH	CHF	瑞士法郎
EU	EUR	欧元
GB	GBP	英镑
HK	HKD	港币
JP	JPY	日元
KW	KRW	韩元
LK	LKR	斯里兰卡卢比
MM	MMK	缅元
NO	NOK	挪威克朗
NZ	NZD	纽西兰币
PO	PO	点数
PP	PHP	菲律宾比绍
RB	RMB	人民币
RM	MYR	马币
RP	IDR	印度尼西亚盾
RS	INR	印度卢比
SE	SEK	瑞典克朗
TB	THB	泰铢
US	USD	美元
VN	VND	越南盾
ZA	ZAR	南非币
ZW	ZWD	津巴布韦币