

JDB Seamless Wallet API 规格书

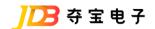
Version: 1.12

Date: 2019-09-18

修订历程

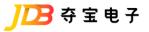
日期	版本	修订项目	修订者
2017-05-31	1.0	i. 创建文件	Brenda Hou
2017-07-07	1.1	i. 取得 token 移除 x, 新增 path	Roger Wu
2018-02-27	1.2	i. 更新 action 11, action 47:	Brenda Hou
		A. 增加验证 gType+mType 是否存在, 验证	
		失败则回传错误代码: 8000	
		B. 新增 isAPP, lobbyURL 参数	
		ii. 新增功能	
		A. 查询游戏列表 Action 49	
		B. 查询使用者是否在游戏中 Action 52	
		C. 查询游戏结果 Action 54	
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	Brenda Hou
		ii. 追加查询单一钱包交易结果	
2018-07-02	1.5	i. 新增功能	Brenda Hou
		A. 赶出用户 Action 17	
		B. 赶出所有下线用户 Action 58	
2018-07-27	1.6	i. 更新 action 21, action 47:	Brenda Hou
		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	
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	Brenda Hou
		ii. 修正 Action 8:gameSeqNo 回传型态由	Amuro Huang
		Integer 改为 Long	
		iii. 用户账号(uid, playerId)长度限制从 String(19)	
		改为 String(25)	
2019-03-21	1.11	i. 加强文件说明	Brenda Hou
		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)	

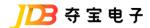
		iv. 范例程序增加 Go Lang 版本	
2019-09-18	1.12	i. 范例程序新增 C# 解密范例	Amuro Huang
		ii. Action 11, 47 新增栏位 isShowDollarSign	



目录

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





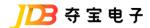
1. 介绍

1.1.注意事项

- ▶ API 网址、加密金錀与游戏列表等信息,请洽本公司业务索取。
- ▶ 使用此系统时,请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
 - ◆ 試玩玩家
 - 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	
х	1.	参阅章节「功能项目」决定要呼叫的功能
	2.	依功能所需的参数转为 JSON String
	3.	将此 JSON String 利用 AES 128 bit 进行解密
	4.	解密步骤,可参考「范例程序」



2.3.功能项目

2.3.1. Action 4:取消下注

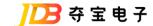
➤ 功能说明 当下注请求(Action 8)失败时呼叫该 API。

▶ 参数

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

Example:

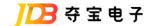
```
{
    "action":4,
    "ts":1447452951820,
    "transferId":"123456789",
    "uid":"testpl01"
}
```



▶ 返回结果

参数	格式	说明
status	String	成功: 0000 错误: 9015:交易序号不存在(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
balance	Double	余额
err_text	String	错误讯息

```
http:Request
http://{ HttpUrl }?x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-UWsfd6PHz9Ky-
wM3mIC9E14kwl_S1X31praVaCLnp-
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

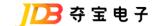
▶ 功能说明 回传玩家剩余金额。

▶ 参数

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

Example:

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



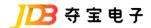
▶ 返回结果

参数	格式	说明
status	String	成功: 0000
		错误: 7501-玩家不存在
		※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
balance	Double	余额
err_text	String	错误讯息

```
Http Request
http://{ HttpUrl }?x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-UWsfd6PHz9Ky-wM3mIC9E14kw1_S1X31praVaCLnp-
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。

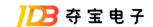
● 常見錯誤代碼說明

✓ 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

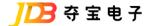
✓ 其他错误代码请参照附录「错误代码」

▶ 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。
gТуре	Integer	游戏型态(请参考「JDB Games.pdf」)
туре	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
hasFreegame	Integer	免费游戏

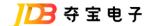


		0: 否
		1: 是
hasGamble	Integer	博取游戏
		0: 否
		1: 是
鱼机下注记录: gType=7		
roomType	Integer	游戏区域
		0:欢乐
		1: 富豪
		2:依机台类型区分
		mType=7003: 财神
		mType=7004: 五龙
街机下注记录: gType=9		
hasGamble	Integer	博取游戏
		0: 否
		1: 是
hasBonusGame	Integer	奖金游戏
		0: 否
		1: 是
电子彩票下注记录: gType	e=12	
hasBonusGame	Integer	奖金游戏
		0: 否
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



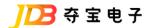
Example - 老虎机:

```
{
   "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
```



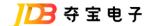
Example - 鱼机:

```
{
   "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
```



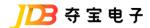
Example - 街机:

```
{
   "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
```



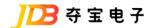
▶ 返回结果

参数	格式	说明
status	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
balance	Double	余额
err_text	String	错误讯息

```
Http Request
http://{ HttpUrl }?x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-UWsfd6PHz9Ky-wM3mIC9E14kw1_S1X31praVaCLnp-
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGy0vsS4v1Yj70vpRw_iU4554RuSsvHp19jaj4XU
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
х	1. 参阅章节「功能项目」决定要呼叫的功能
	2. 依功能所需的参数转为 JSON String
	3. 将此 JSON String 利用 AES 128 bit 进行加密
	4. 加密步骤,可参考「范例程序」

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

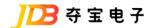
- ▶ 功能说明
 - 如果玩家在线,玩家将会收到一封被赶出的提示讯息。
 - 如果玩家正在进行游戏,游戏将会被中止并直接结算。

▶ 参数

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

Example:

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



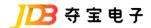
▶ 返回结果

参数	格式	说明
status	String(4)	成功: 0000
		错误:参照附录「错误代码」
err_text	String(255)	错误讯息

```
http Request
http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-
UWsfd6PHz9Ky-wM3mIC9E14kw1_S1X31praVaCLnp-
K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XU
gTK5yzbWKEddANjAAbxF0s=

Http Response (success)
{
    "status":"0000"
}

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

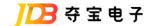


3.2.2. Action 21:取得 Token

▶ 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
action	Integer	Υ	21
ts	Long	Υ	当前系统时间
parent	String(19)	Υ	代理账号
uid	String(25)	Υ	玩家账号
balance	Double	Υ	余额
lang	String(2)	N	语系
			en: 英文(默认值)
			ch: 简体中文
gType	String(1)	N	游戏型态
mType	String(3)	N	机台类型
windowMode	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。
isAPP	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)
lobbyURL	String(1000)	N	游戏大厅网址
			当 windowMode 为 2 时,此参数才会有作用
moreGame	Integer	N	0: 不显示更多游戏
			1: 显示更多游戏(默认值)
mute	Integer	N	默认音效开关
			0: 开启音效(默认值)
to the section	1.1		1: 静音
jackpotFlag	Integer	N	设定彩金开关
			0: 打开(默认值)
isola B. II or	D I.		1: 关闭
isShowDollarSign	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

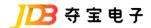
Example:

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

返回结果

参数	格式	说明
status	String(4)	成功: 0000
		错误:参照附录「错误代码」
err_text	String(255)	错误讯息
path	String(255)	开启游戏连结

```
Http Request
  http://{API_Address}/apiRequest.do?dc=jb1&x=y1wayxScczy6ABfElfaVwb5CdfHxBN_JGtxyBS
  vpohLQakP8M2QSetJNojJBzvnrVChhq8YWWgBTdGeXFTH4vSNQQ66epf4NqYeQ-
  6bJ15pvHctQjQevulv7GHqOifuju gY0H54FfVdQc9D5tKh1AZ-
  9KDh9dwcGSszZxU7JOOoHdYITlCO5LhvGmHxPBwZ
Http Response (success)
 {
    "status":"0000",
    "path":"http://xxx.xxx?x=Q0Y2NUZDNEFBODEzNUM5NURGRjc5QTUzRDk0R
    T1DRDUzNEQ0MjhEOEMzMURDODBCNzdCMEE2OTE3MzczRkE5RDNGQ0QzQjdFNT1B
    MjE4Njg5NDM4RDREMEI1NkYzQjkxOEU5QTU0QTM5QTY1MEUyNjYzMDg1RkFBQTM
    10UU3RTI2RTBGQUEzMUI0MkQ4NEFBRjVDNkJFRkQzMzU1NzA50UNDMkYxMzMyOT
    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

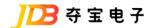
- 查询单一代理在特定日期内的游戏详细交易信息
- 查询日期为 01-01-2016,则取得数据范围为 01-01-2016 12:00:00 至 01-02-2016 12:00:00。
- 每次查询最新信息为1天前的资料。
 - ◆ 例 1: 系统时间为 05-01-2016 18:23:00, 则查询日期需小于等于 04-01-2016
 - ◆ 例 2: 系统时间为 05-01-2016 09:23:02,则查询日期需小于等于 03-01-2016 (因 04-01-2016 的报表日期还在进行中)
 - ◆ 例 3. 系统时间为 05-01-2016 12:11:22, 则查询日期需小于等于 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
action	Integer	Υ	42
ts	Long	Υ	当前系统时间
parent	String(19)	Υ	代理账号
gТуре	String(1)	Υ	游戏型态
date	String(10)	Υ	查询日期(dd-MM-yyyy)

Example:

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

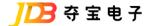


▶ 返回结果

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

操作范例

```
Http Request
  http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-
  UWsfd6PHz9Ky-wM3mIC9E14kwl_S1X31praVaCLnp-
  KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XU
  gTK5yzbWKEddANjAAbxF0s=
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

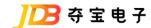
➤ 功能说明 针对部份玩家关闭 Jackpot 的功能

参数

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

Example:

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



▶ 返回结果

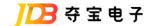
参数	格式	说明
status	String(4)	成功: 0000
		错误: 参照附录「错误代码」
err_text	String(255)	错误讯息

```
Http Request

http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-UWsfd6PHz9Ky-wM3mIC9E14kwl_S1X31praVaCLnp-
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
action	Integer	Υ	45
ts	Long	Υ	当前系统时间
parent	String(19)	Υ	代理账号

Example:

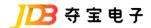
```
{
    "action":45,
    "ts":1447452951820,
    "parent":"testag"
}
```

▶ 返回结果

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



```
Http Request
  http://{API Address}/apiRequest.do?dc=jb&x=udbFVt xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-
  UWsfd6PHz9Ky-wM3mIC9E14kwl_S1X31praVaCLnp-
  KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_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."
  }
```

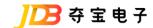


3.2.6. Action 47:取得试玩 Token

➤ 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
action	Integer	Υ	47
ts	Long	Υ	当前系统时间
lang	String(2)	N	语系
			en: 英文(默认值)
			ch: 简体中文
gТуре	String(2)	N	游戏型态
туре	String(5)	N	机台类型
			※若需直接进入游戏,需同时带入 mType 及
			gType
windowMode	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, 隐藏游戏中的关闭钮
			※gType 及 mType 为必填字段。
isAPP	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)
lobbyURL	String(1000)	N	游戏大厅网址
_			当 windowMode 为 2 时, 此参数才会有作用
moreGame	Integer	N	0: 不显示更多游戏
			1: 显示更多游戏(默认值)
mute	Integer	N	默认音效开关
			0: 开启音效(默认值)
ischourDollows:	Dooloon	NI NI	1: 静音
isShowDollarSign	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
			Igise: 小水小山如小 A



Example:

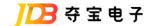
```
{
    "action":47,
    "ts":1447490495704,
    "lang":"ch",
    "gType":"0",
    "mType":"8001"
    "windowMode":"2",
    "lobbyURL":"http://xxx.xxx.xxx/lobby.html"
}
```

▶ 返回结果

参数	格式	说明	
status	String(4)	成功: 0000	
		错误:参照附录「错误代码」	
err_text	String(255)	错误讯息	
path	String(255)	登入 URL	

操作范例

```
Http Request
  http://{API_Address}/apiRequest.do?dc=jb1&x=y1wayxScczy6ABfElfaVwb5CdfHxBN_JGtxyBS
  vpohLQakP8M2QSetJNojJBzvnrVChhq8YWWgBTdGeXFTH4vSNQQ66epf4NqYeQ-
  6bJ15pvHctQjQevulv7GHqOifuju gY0H54FfVdQc9D5tKh1AZ-
  9KDh9dwcGSszZxU7JOOoHdYITlCO5LhvGmHxPBwZ
Http Response (success)
 {
    "status":"0000",
    "path":"http://xxx.xxx.xxx/?x=Q0Y2NUZDNEFBODEzNUM5NURGRjc5QTUzRDk0
    RT1DRDUzNEQ0MjhEOEMzMURDODBCNzdCMEE2OTE3MzczRkE5RDNGQ0QzQjdFNT1
    BMjE4Njg5NDM4RDREMEI1NkYzQjkxOEU5QTU0QTM5QTY1MEUyNjYzMDg1RkFBQT
    M10UU3RTI2RTBGQUEzMUI0MkQ4NEFBRjVDNkJFRkQzMzU1NzA50UNDMkYxMzMyO
    ThBNjhBRURCNzExREMzRjhDNzA1N0VFNEIzMzhCNTFEQ0FDODVEODQ3NEFDNTE1
    MEE1NTMzODQ="
 }
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
action	Integer	Υ	49
ts	Long	Υ	当前系统时间
parent	String(19)	Υ	代理账号
lang	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
status	String(4)	成功: 0000
		错误:参照附录「错误代码」
data	JSON array	
gType	Integer	游戏型态
list	JSON array	
mType	Integer	机台类型
isNew	Boolean	是否为新游戏
image	String	游戏图片网址
name	String	游戏名称
err_text	String(255)	错误讯息



操作范例

```
Http Request
  http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-
  UWsfd6PHz9Ky-wM3mIC9El4kwl_SlX3lpraVaCLnp-
  KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_iU4554RuSsvHpI9jaj4XU
  gTK5yzbWKEddANjAAbxF0s=
Http Response (success)
  {
       "status":"0000",
       "data":[
               "gType": 0,
               "list": [
                         "mType": 8001,
                         "isNew": false,
                         "image": "https://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

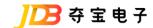
参数	格式	强制	说明
action	Integer	Υ	52
ts	Long	Υ	当前系统时间
parent	String(19)	Υ	代理账号
uid	String(25)	Y	玩家账号

Example:

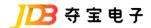
```
{
    "action":52,
    "ts":1447452951820,
    "parent":"testag",
    "uid":" testpl01"
}
```

▶ 返回结果

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



```
Http Request
           http://\{API\_Address\}/apiRequest.do?dc=jb\&x=udbFVt\_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-thtp://thtps://thesamlong.doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc-policy-doc
           UWsfd6PHz9Ky-wM3mIC9El4kwl_SlX3lpraVaCLnp-
           KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_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

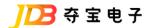
- > 功能说明
 - 查询玩家游戏结果
 - 未带入 gType 与 seqNo 会进入报表主页
 - 有带入 gType 与 seqNo 会直接进入该次游戏的详细信息页面

参数

参数	格式	强制	说明
action	Integer	Υ	54
ts	Long	Υ	当前系统时间
parent	String(19)	Υ	代理账号
uid	String(25)	Υ	玩家账号
lang	Stirng(5)	N	en: 英文 (默认值)
			ch: 简体中文
gТуре	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)

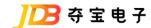
Example:

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



▶ 返回结果

参数	格式	说明
status	String(4)	成功: 0000
		错误:参照附录「错误代码」
data	JSON array	
path	String	游戏结果 URL
err_text	String(255)	错误讯息



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)

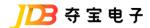
若 parent 为 agent 账号,则此 agent 账号下所有会员皆会被赶出系统。

▶ 参数

参数	格式	强制	说明
action	Integer	Υ	58
ts	Long	Υ	当前系统时间
parent	String(19)	Υ	上线账号

Example:

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



▶ 返回结果

参数	格式	说明
status	String(4)	成功: 0000
		错误:参照附录「错误代码」
err_text	String(255)	错误讯息

```
http Request
  http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-
UWsfd6PHz9Ky-wM3mIC9E14kwl_S1X31praVaCLnp-
  K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_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

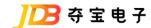
- 查询在特定时间内的单一钱包事务历史记录
- 开始时间与结束时间中,ss(秒数)的值必须为00
- 每次查询时间范围最多为 15 分钟,最新信息为 3 分钟前。例如,在 15:00:00 时的查询,最晚只能查到 14:56:00~14:57:00 的信息。
- 建议每次查询范围不超过1分钟,以获得较佳效果。
- 提供2小时内的交易结果。

▶ 参数

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

Example:

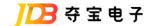
```
{
    "action":59,
    "ts":1447452951820,
    "starttime":"08-12-2016 17:45:00",
    "endtime":"08-12-2016 17:46:00"
}
```



▶ 返回结果

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

```
Http Request
  http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-
  UWsfd6PHz9Ky-wM3mIC9E14kwl_S1X31praVaCLnp-
  K0WsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_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

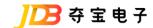
参数	格式	强制	说明
action	Integer	Υ	62
ts	Long	Υ	当前系统时间
startDate	String(10)	Υ	查询区段起始日期 (dd-MM-yyyy)
endDate	String(10)	Υ	查询区段结束日期 (dd-MM-yyyy)

Example:

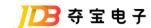
```
"action":62,
"ts":1447452951820,
"startDate":"29-08-2018",
"endDate":"29-08-2018"
}
```

▶ 返回结果

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



```
Http Request
  http://{API_Address}/apiRequest.do?dc=jb&x=udbFVt_xv0tsAmLDpz5Z3Ct4-p0gedUPdQ0-
  UWsfd6PHz9Ky-wM3mIC9El4kwl SlX3lpraVaCLnp-
  KOWsgKmpYTV9XpYncHzbtvn591qfaAwpGyOvsS4v1Yj7OvpRw_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"}
                 1
              },{
                 "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."
```



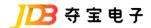
4. 范例程序

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.lvParameterSpec;
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);
    }
}
```

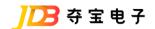


▶ 解密

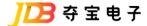
```
public static String decrypt(String data, String key, String iv) throws Exception {
    Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding");
    cipher.init(Cipher.DECRYPT_MODE, new SecretKeySpec(key.getBytes(), "AES"),
    new IvParameterSpec(iv.getBytes());
    String plainText = new String(cipher.doFinal(Base64.decodeBase64(encData)));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

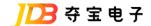


```
public class sample {
     public static JsonObject jsonObjectencryptAndQuery(JsonObject v, boolean zipResponse)
throws Exception {
         JsonObject rJson = null;
         String x;
         // get a client
         CloseableHttpClient demo = HttpClientBuilder.create().build();
         String 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

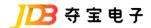
```
@Test
public void testPerformAction21() throws Exception {
      // prepare data
      JsonObject v = generateNewVector();
      v.addProperty("action", 21);
      v.addProperty("ts", $ {system time});
      v.addProperty("parent", $ {parent id});
      v.addProperty("uid", $ {player id});
     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

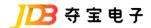
- ▶ 说明:
 - 以 php7.0 程序、登入系统交易信息为例
- ➤ AES 加密、解密 (MCrypt.php)

```
<?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 = ' ';
         $size
                       = 16;
                         = strlen($source) % $size;
         $x
         $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>";
?>
```



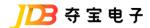
4.3.C#

- ▶ 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.Ling;
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 ison and use the path url to login game
         JObject obj = JObject.Parse(response.Content);
         String path = (String) obj["path"];
         System.Console.WriteLine(path);
```

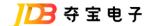


```
public static string AESEncryptToString(object rawData) {
         return AESEncryptToString(JsonConvert.SerializeObject(rawData));
     }
     public static string AESEncryptToString(string JsonString) {
         var aESEncryptBytes = AESEncrypt(JsonString);
         var URL_Safe_AESEncrypt_String = 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;
         }
     }
}
```

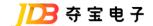


➤ AES 解密

```
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();
```



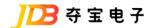
```
decrypt = Encoding.UTF8.GetString(ms.ToArray());
}
}
catch (Exception e)
{
    Console.WriteLine(e.Message);
}
return decrypt;
}
```



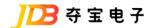
4.4.Go Lang

- ▶ 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

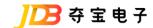
```
package main
import (
     "bytes"
      "crypto/aes"
      "crypto/cipher"
      "encoding/base64"
      "errors"
     "fmt"
     "time"
     "strings"
     "strconv"
      "net/http"
      "io/ioutil"
)
const aesKeyDefValue = ${AES KEY}; // change it
const aesIvDefValue = ${AES IV}; // change it
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)
     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
}
```



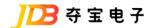
```
func AesDecrypt(ciphertext []byte, key []byte) ([]byte, error) {
     block, err := aes.NewCipher(key)
     if err != nil {
          return nil, errors.New("invalid decrypt key")
     blockSize := block.BlockSize()
     if len(ciphertext) < blockSize {</pre>
          return nil, errors.New("ciphertext too short")
     iv := []bvte(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)]
```



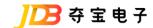
5. 附录

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.



7405	You have been logged out!
7501	User ID cannot be found.
7502	User is suspended.
7503	User is locked.
7504	User is closed.
7505	User is not playing
7601	Invalid User ID. Please only use characters between a-z, 0-9
7602	Account already exist. Please choose other User ID
7603	Invalid username.
7604	The password must at least 6 characters, with 1 alphabet and 1 number.
7605	Invalid operation_code. Please only use number 2, 3, 4, 5.
6001	Your Cash Balance not enough to withdraw
6002	User balance is zero
6003	Withdraw negative amount
6004	Duplicate Transfer
6005	Repeat serial number.
6006	Your Cash Balance not enough.
6901	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
ВТ	BDT	孟加拉国塔卡
CA	CAD	加币
СН	CHF	瑞士法郎
EU	EUR	欧元
GB	GBP	英镑
НК	HKD	港币
JP	JPY	日元
KW	KRW	韩元
LK	LKR	斯里兰卡卢比
MM	ММК	缅元
NO	NOK	挪威克朗
NZ	NZD	纽西兰币
РО	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
ТВ	ТНВ	泰铢
US	USD	美元
VN	VND	越南盾
ZA	ZAR	南非币
ZW	ZWD	津巴布韦币