旗舰厅平台 API 文档

版本记录

日期	版本	描述	作者
01/25/2018	V1.0	版本创建	POD
04/24/2018	V1.1	增加接口功能，参数的详细说明	POD
05/21/2018	V1.2	补充加密算法示例	POD
12/11/2019	V1.2	补充参数说明	POD

目　录

旗舰厅平台 API 文档 1

1　说明 4

2　API 接口说明 6

2.1　流程图 6

2.1.1　API 基本结构 6

2.1.2　注册新账号 7

2.1.3　查询账户结余 8

2.1.4　转账流程 8

2.2　API 接口详情 10

2.2.1　用户登录 10

2.2.2　验证并创建账号 11

2.2.3　查询余额 11

2.2.4　预转账 12

2.2.5　转账确认 12

2.2.6　转账记录查询 13

2.2.7　启用/禁用账户 14

2.2.8　修改账户密码 14

2.2.9　用户进入游戏【接口地址是 GCI_WEB】 15

2.2.10　厅方回调网站[游戏客户触发] 15

2.3　网站进入旗舰厅游戏 16

2.3.1　流程图 16

3 附录 17
3.1 接口返回码 17
3.2 APP 与 MH5 交互接口 19
3.3 APP 内嵌 MH5 传参 19
3.4 参数值参考说明 20
3.4.1 webApp，inApp 参数说明 20
3.4.2 lang 参数说明 20
3.4.3 gameType,videoID 参数说明 20
3.5 加密串校验示例 23
3.6 Java 加密算法示例 23
3.7 Node JS 加密算法示例 25
3.8 PHP 加密算法示例 25
3.9 C++加密算法示例 29
4 Q&A 29

1 说明

此文档仅用于对第三方网站接入旗舰厅的 API 说明和指导文档。

接口调用规则：

- 采用 POST 方法
- 请求头 content-type=application/xml
- agent 参数【代理商名称】，通过请求 url 的 query 参数传递。目前会提供真钱和试玩两种代理线，注意在接入过程中严格区分，不能混用。目前试玩账号不需要调用转账接口，厅方默认会给进厅的试玩账号重置 2000 的额度，同时也建议网站方创建试玩账号池 100-500 左右，循环利用此试玩账号，以免在厅方创建过多的意义不大的试玩账号。完整代理线信息请参考提供的文档【旗舰厅_AgentInfo_4_产品.xlsx】
- params 参数，所有参数以 Json 形式加密后，通过 body 传递
- 响应头: application/xml;charset=UTF-8
- Params 加密算法模板参照【附录】

加密算法**(AES/ECB/PKCS5Padding)** 如下：

---

key = MD5(authPwd)//authPwd 代理密码

params = AES128^{key}(Json)

请求格式样例如下：

请求地址	http://GameGI.Domain/xxx**?agent=xxx**	
描述	接口请求样例	
请求方式	POST	
请求头	content-type=application/xml	
请求参数 [params]	3780abe47b0e28277cce160e2465e75aa1f8337d80ab5490e644e54f7d807d	参数密文

响应头	application/xml;charset=UTF-8	
响应数据	<response xmlns=""> <code>0</code> <message>success</message> <body/> </response>	
说明	code	响应码
	message	提示信息
	body	响应体

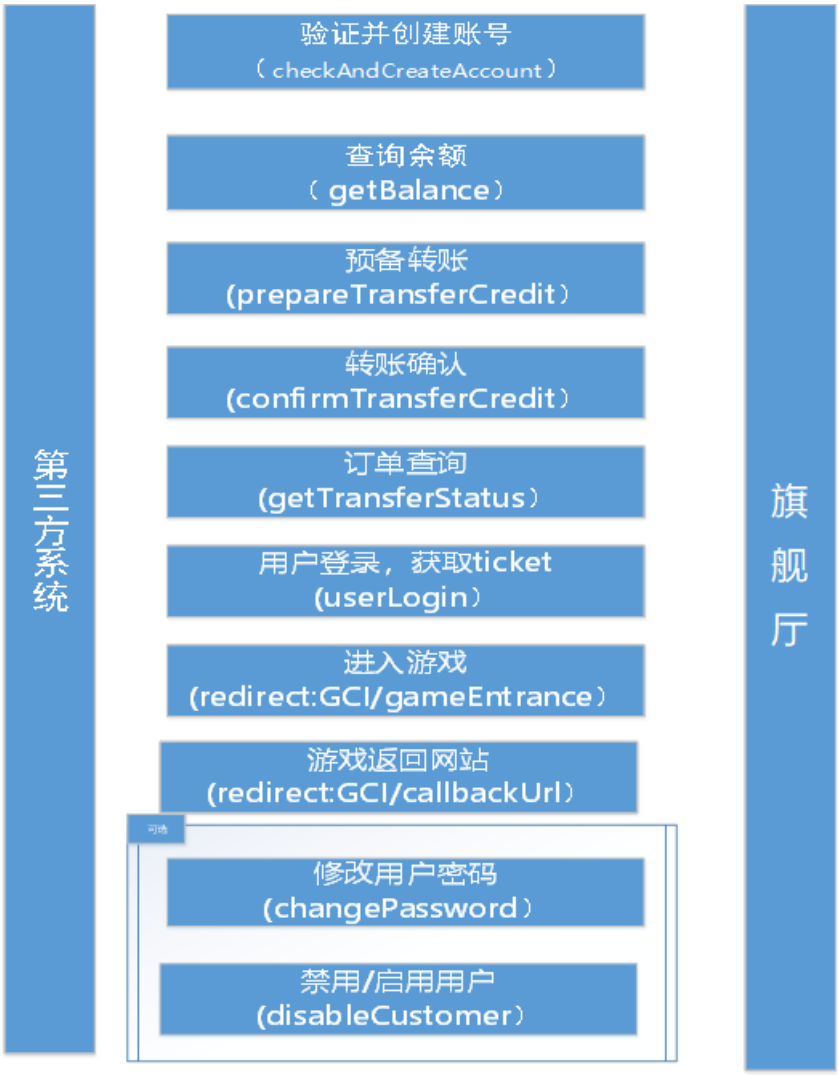
业务相关的说明：

- 目前游戏支持的语言类型：中文(ZH)，英文(EN)，越南文(VI)，日文(JP)
- 游戏中所有数据的时间为美东时间【与北京时间相差 12 小时】
- 所有接口需要的时间戳都是精确到毫秒，用于厅方进行接口的请求时间校验，默认超过 5 分钟接口调用就判断为超时【例外：2.2.9 用户进入游戏 ticket 中的时间戳默认 30S-可配】。
- 厅方转账目前都只支持整数部分的转入和转出，详情可咨询运营具体业务。

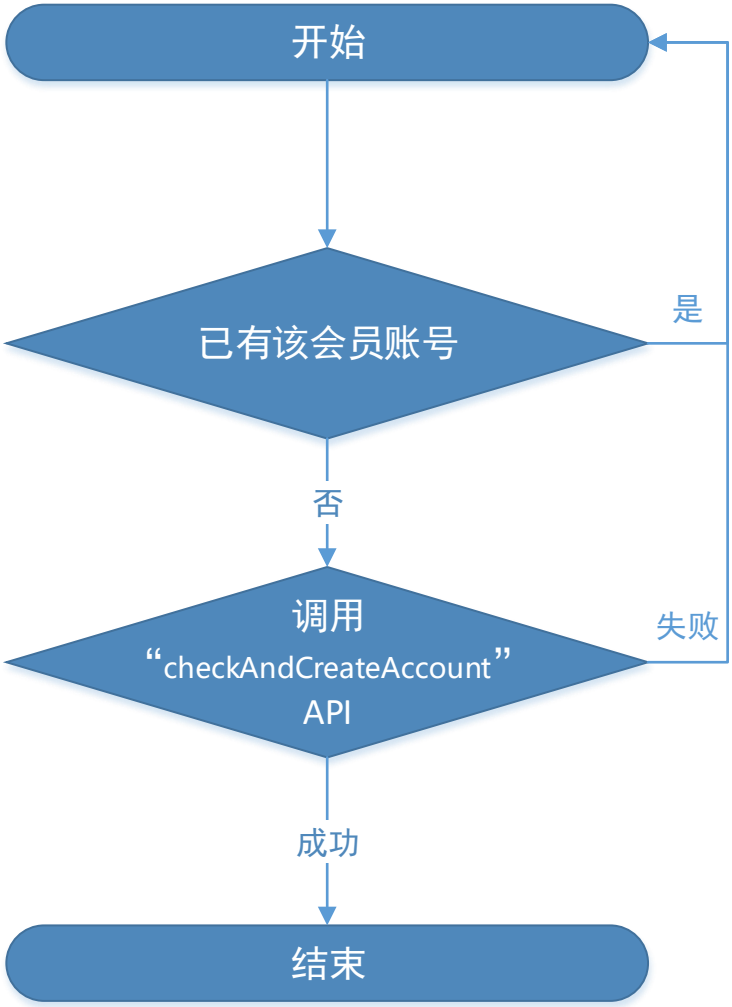
2 API 接口说明

2.1 流程图

2.1.1 API 基本结构



2.1.2 注册新账号

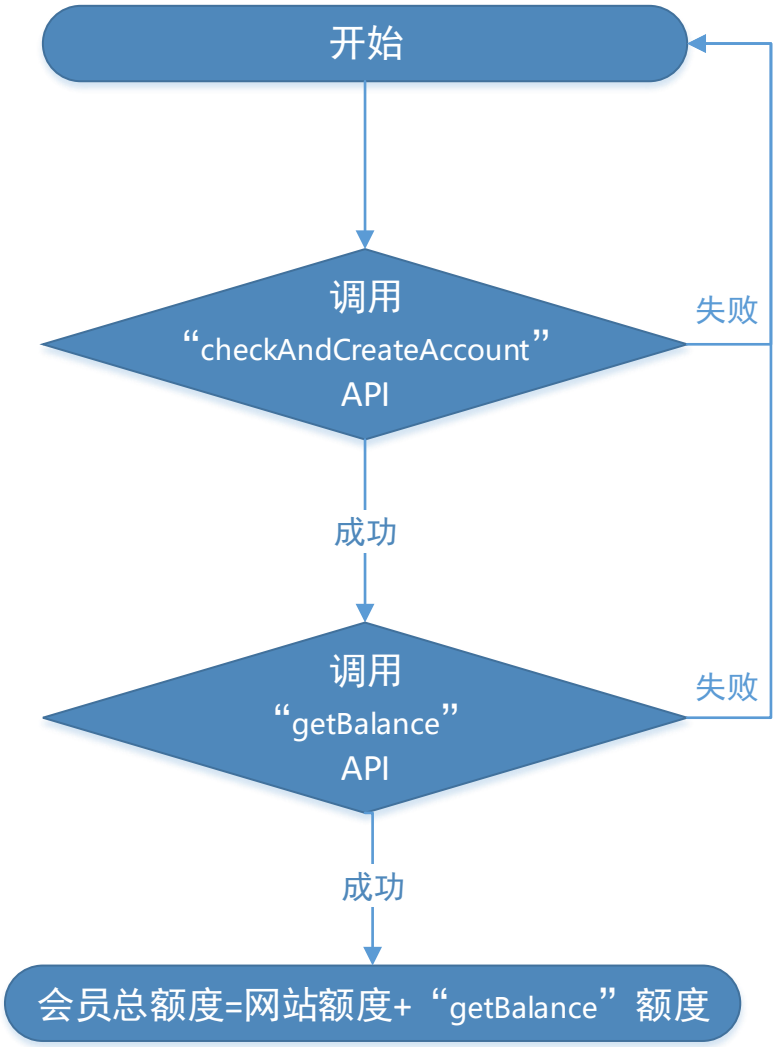


说明：

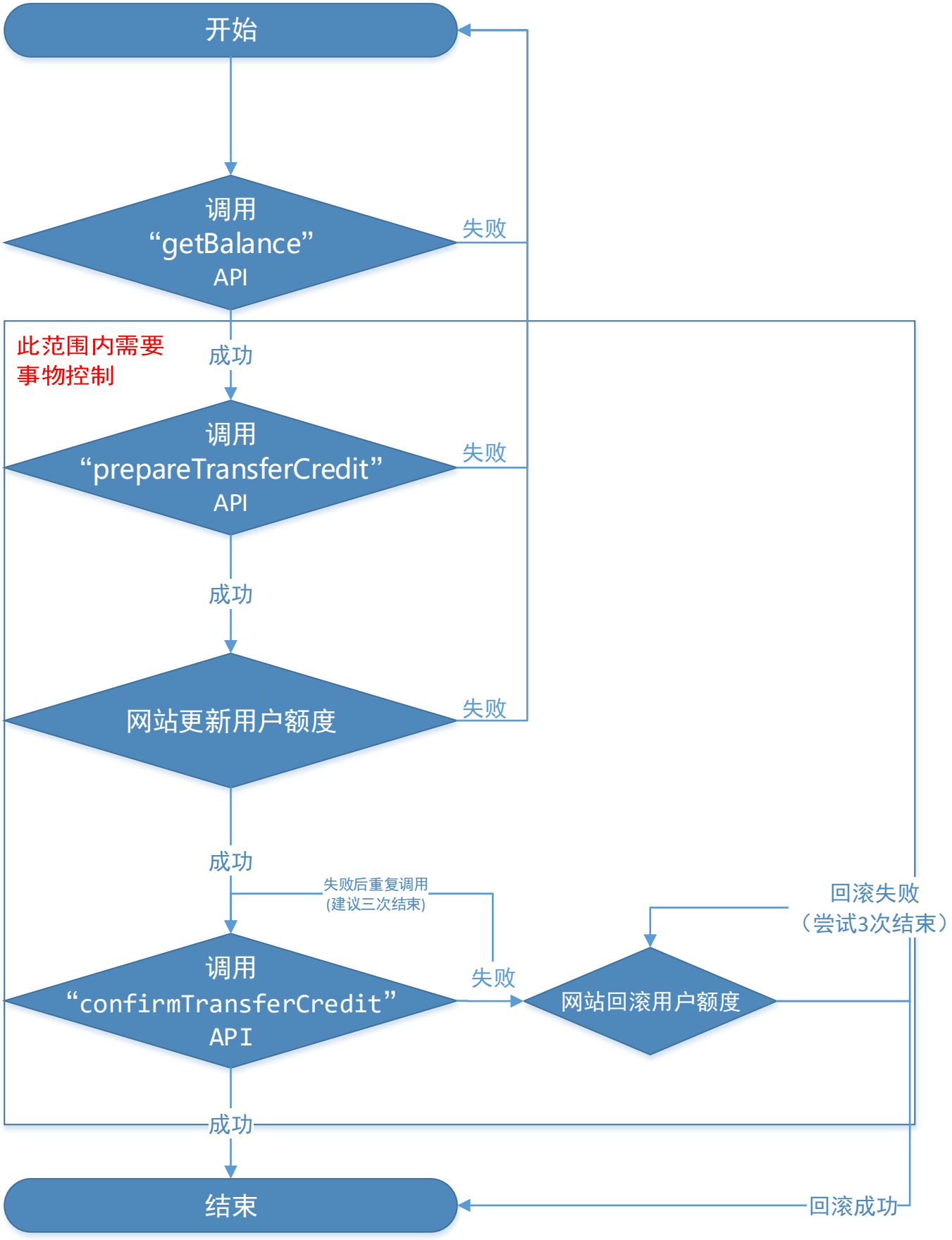
1 盘口 oddType 请设置成固定值 A，目前该字段属于保留字段.

2 试玩账号额度默认为 2000，每次登陆厅方都会生成唯一账号

2.1.3 查询账户结余



2.1.4 转账流程


开始
调用
“getBalance”
API
失败
此范围内需要
事物控制
成功
调用
“prepareTransferCredit”
API
失败
成功
网站更新用户额度
失败
成功
失败后重复调用
(建议三次结束)
回滚失败
（尝试3次结束）
调用
“confirmTransferCredit”
API
失败
网站回滚用户额度
成功
结束
回滚成功

说明：

1 为了保证额度的一致性，建议上面红色字体方框范围里的接口调用需要事物控制

2 确认转账接口(confirmTransferCredit),调用失败，如果查询转账记录接口没有记录，可以再次用该 billno 重复调用.

2.2 API 接口详情

API 相关接口测试地址:

http://GameGI.Domain/swagger-ui.html

2.2.1 用户登录

请求地址	http://GameGI.Domain/userLogin?agent=xxx			
描述	用于进入游戏			
请求方式	POST			
请求头	content-type=application/xml			
加密参数 [params]	字段名	类型	必须	备注
	timestamp	String	Y	时间戳(毫秒)
	userName	String	Y	用户名
	password	String	Y	用户密码(32 位 MD5 值)
	tradeCurrency	String	N	用户交易货币类型：CNY,USDT,PHP,【例：人民币线路用 1：7 usdt 换算方式进入厅方，此处交易货币类型应该传 USDT】，默认是用户本代理线对于币种
响应头	application/xml;charset=UTF-8			
响应数据	<response xmlns=""> <code>0</code> <message>success</message> <body> <ticket>xxxxx</ticket> <session>xxxxxxx< session > </body> </response>			
说明	code			响应码[非零即为失败]
	message			提示信息
	body			响应体
	ticket			加密票据。进入游戏调用 gameEntrance 接口时需要此 ticket 参数值

2.2.2 验证并创建账号

请求地址	http://GameGI.Domain/checkAndCreateAccount?**agent=xxx**			
描述	验证并创建账号			
请求方式	POST			
请求头	content-type=application/xml			
加密参数 [params]	字段名	类型	必须	备注
	timestamp	String	Y	时间戳(毫秒)
	userName	String	Y	用户名
	password	String	Y	用户密码(32 位 MD5 值)
	aliasName	String	N	不传则默认为 userName 的值
	oddType	String	N	备用字段，暂可以不传值
	currency	String	N	备用字段，暂可以不传值
响应头	application/xml;charset=UTF-8			
响应数据	<response xmlns=""> <code>0</code> <message>success</message> <body/> </response>			
说明	code			响应码[非零即失败]
	message			提示信息
	body			响应体

2.2.3 查询余额

请求地址	http://GameGI.Domain/getBalance?**agent=xxx**			
描述	查询账户余额			
请求方式	POST			
请求头	content-type=application/xml			
加密参数 [params]	字段名	类型	必须	备注
	timestamp	String	Y	时间戳(毫秒)
	userName	String	Y	用户名
响应头	application/xml;charset=UTF-8			

响应数据	<response xmlns=""> <code>0</code> <message>success</message> <body> <productId>A01</productId> <userName>mina1234561</userName> <createDate>2017-11-21 2:07:52</createDate>//用户创建时间【美东时间】 <disable>N</disable>//用户是否被禁用 <currency>CNY</currency>//货币类型 <parentName>012001001001006</parentName>//该用户的上级代理 agcode <balance>1000000</balance>//额度 【目前只支持查询的都是正整数】 </body> </response>	
说明	code	响应码[非零即失败]
	message	提示信息
	body	响应体

2.2.4 预转账

请求地址	http://GameGI.Domain/prepareTransferCredit?**agent=xxx**			
描述	预转账			
请求方式	POST			
请求头	content-type=application/xml			
加密参数 [params]	字段名	类型	必须	备注
	timestamp	String	Y	时间戳(毫秒)
	userName	String	Y	用户名
	billno	String	Y	订单号[保证唯一性]
	action	String	Y	操作类型，IN ： 存入 OUT ： 取出
	credit	String	Y	转账额度[必须大于 0 的正整数，暂不支持小数]
响应头	application/xml;charset=UTF-8			
响应数据	<response xmlns=""> <code>0</code> <message>success</message> <body/> </response>			
说明	code			响应码[非零即失败]
	message			提示信息
	body			响应体

2.2.5 转账确认

请求地址	http://GameGI.Domain/confirmTransferCredit?**agent=xxx**

描述	确认转账			
请求方式	POST			
请求头	content-type=application/xml			
加密参数 [params]	字段名	类型	必须	备注
	timestamp	String	Y	时间戳(毫秒)
	userName	String	Y	用户名
	billno	String	Y	订单号[保证唯一性,必须和预转账的 billno 一样]
	action	String	Y	操作类型，IN ： 存入　OUT ： 转出
	credit	String	Y	转账额度[必须大于 0 的正整数，暂不支持小数]
响应头	application/xml;charset=UTF-8			
响应数据	<response xmlns=""> <code>0</code> <message>success</message> <body/> </response>			
说明	code			响应码[非零即失败]
	message			提示信息
	body			响应体

2.2.6 转账记录查询

请求地址	http://GameGI.Domain/getTransferStatus?**agent=xxx**			
描述	转账状态查询			
请求方式	POST			
请求头	content-type=application/xml			
加密参数 [params]	字段名	类型	必须	备注
	timestamp	String	Y	时间戳(毫秒)
	billno	String	Y	订单号[必须和预转账的 billno 一样]
响应头	application/xml;charset=UTF-8			
响应数据	<response xmlns=""> <code>0</code> <message>success</message> <body> <productId>A01</productId> <userName>mina1234561</userName> <createTime>2018-01-15 02:52:14</createTime>//创建时间【美东时间】 <transType>IN</transType>//转账类型 <amount>1</amount>//转账额度 <balance>1000004</balance>//转账后额度 <transId>1515999087392</transId>//转账单号 <flag>success</flag>//转账结果 </body> </response>			

说明	code	响应码[非零即失败]
	message	提示信息
	body	响应体
	flag	success:成功，failed:失败

2.2.7 启用/禁用账户

请求地址	http://GameGI.Domain/disableCustomer?agent=xxx			
描述	禁用\启用用户账号. **禁用账号时，如果该账号在游戏中会被自动退出游戏**			
请求方式	POST			
请求头	content-type=application/xml			
加密参数 [params]	字段名	类型	必须	备注
	timestamp	String	Y	时间戳(毫秒)
	userName	String	Y	用户名
	disable	String	Y	0 启用，1 禁用
响应头	application/xml;charset=UTF-8			
响应数据	<response xmlns=""> <code>0</code> <message>success</message> <body/> </response>			
说明	code			响应码[非零即失败]
	message			提示信息
	body			响应体

2.2.8 修改账户密码

请求地址	http://GameGI.Domain/changePassword?agent=xxx			
描述	修改用户密码			
请求方式	POST			
请求头	content-type=application/xml			
加密参数 [params]	字段名	类型	必须	备注
	timestamp	String	Y	时间戳(毫秒)
	userName	String	Y	用户名
	password	String	Y	新的密码(32 位 MD5 值)
响应头	application/xml;charset=UTF-8			
响应数据	<response xmlns=""> <code>0</code> <message>success</message> <body/> </response>			

说明	code	响应码[非零即失败]
	message	提示信息
	body	响应体

2.2.9 用户进入游戏【接口地址是 GCI_WEB】

请求地址	http://GCI_WEB.Domain/gameEntrance?**agent=xxx&type=xx&callbackUrl=&ticket=xxx**			
描述	用于进入游戏			
请求方式	GET			
参数	字段名	类型	必须	备注
	agent	String	Y	代理商名称
	type	String	N	客服端类型[h5]，不传默认进入 pch5
	callbackUrl	String	Y	游戏回调地址 [URL 要带上协议号：http,https]
	lang	String	N	语言类型[ZH,EN,JA,VI] [参考说明]
	gameType	String	N	游戏类型[参考说明]
	videoID	String	N	视频 ID[参考说明]
	ticket	String	Y	用户票据
响应头	application/xml;charset=UTF-8			
响应	重定向到游戏页面			
说明	type			游戏类型参数：h5，不传默认进入 pch5
	gameType,videoID			参数组合可进入指定游戏[参考说明]
	ticket			请求 GameGI userLogin 接口服务返回的票据信息，里面超时时间默认 30S【可配】
	callbackUrl			若三方网站不传此参数，取默认厅方配置 [URL 要带上协议号：http,https]
	APP 内嵌 MH5 专项说明: webApp【2 选 1】 inApp【2 选 1】			APP 与 MH5 交互接口接入[参考说明] APP 内嵌 MH5 传参[参考说明] 使用情况见[场景传参说明]

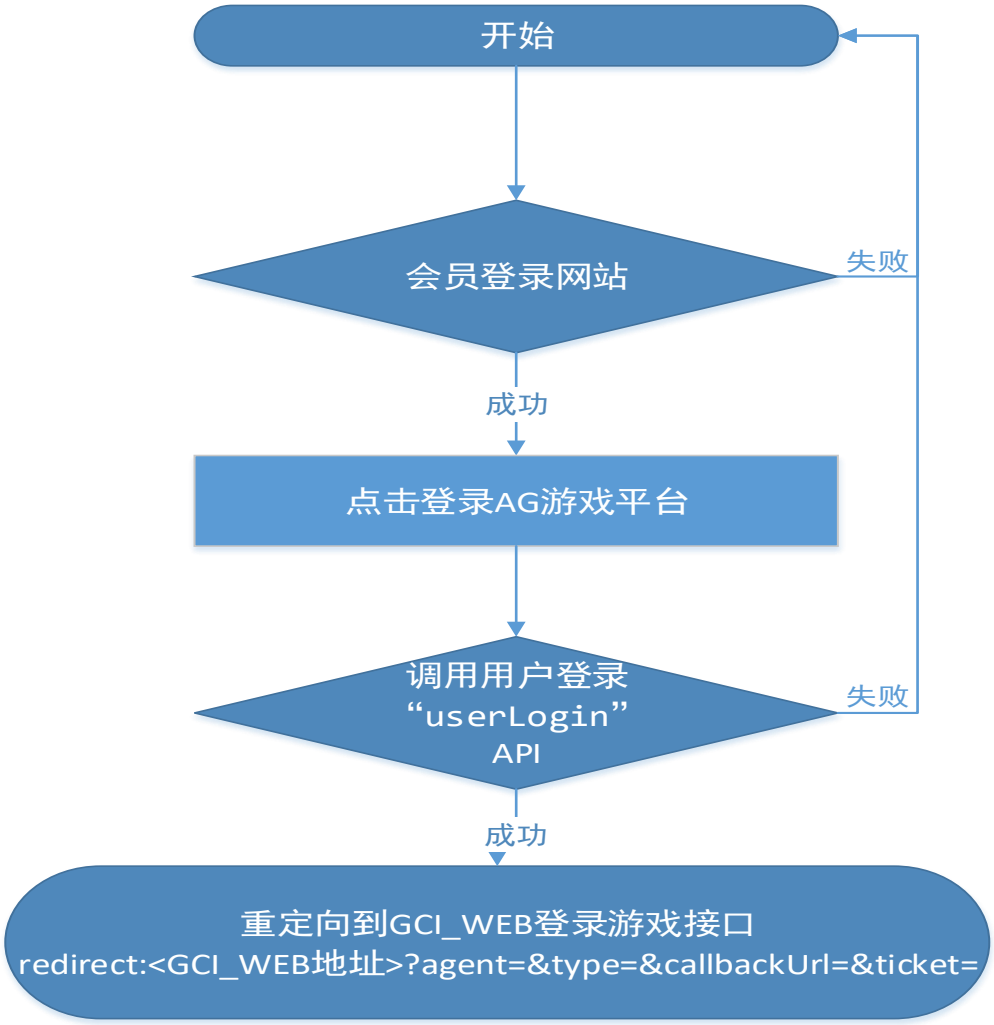
2.2.10 厅方回调网站[游戏客户触发]

请求地址	**callbackUrl**?method=xx&userName=xx&timestamp=xx&key=xx
描述	验证并创建账号

请求方式	GET			
加密参数 [params]	字段名	类型	必须	备注
	method	String	Y	回调类型 rg：开户 dp：存款 wd：取款 et：退出游戏 cs：客服[flash] pcs:客服[h5]
	userName	String	Y	用户名
	timestamp	String	Y	时间戳【毫秒】
	key	String	Y	认证 key
说明	String key = CipherUtils.string2MD5CipherText(method + userName + timestamp); 重定向到指定的 url,可以用 key 校验是否从游戏端返回			

2.3 网站进入旗舰厅游戏

2.3.1 流程图



- 首先网站接入方需要申请旗舰厅的接入代理线，由厅方提供代理账号，密码和产品 ID。
- 厅方需要提供给接入方 GameGI，GCI_WEB 的访问地址。
- 进入游戏和会员转账建议用异步方式实现，以提高用户进入游戏的速度.

3 附录

3.1 接口返回码

code	message	备注
0	success	成功
1	unknown error	未知错误
2	key error	key 不符合
3	error code	DB 错误

4	no userName	没有用户名
5	no password	没有密码
6	no productId	没有产品编号
7	no resetCredit	没有充值额度标识字段
8	no authPwd	没有认证密码
9	no oddType	没有盘口
10	no agent	没有代理名称
11	no billno	没有订单号
12	no action	没有转账类型
13	no credit	没有转账额度
14	no disable	没有禁用标识
15	invalid timestamp	无效的时间戳
16	password not match	密码不匹配
17	agent not exist	代理不存在
18	customer not exist	用户不存在
19	invalid action	无效的转账类型
20	invalid credit	无效的额度
21	invalid disable	无效的禁用标识
22	not found data of billno	没有该笔订单的数据
23	customer is disabled	用户已被禁用
24	no post data	post 请求没有参数
25	agent and productId not match	代理和产品 id 不匹配
26	is not 5 level agent	不是五级代理
27	exist customers	用户已存在
28	is diffents customers agcode	代理号和玩家自身代理号不匹配
29	invalid parameter or parameter is null	请求参数验证失败
30	agent no have credit	代理额度不足
31	invalid agent credi	验证修改代理额度错误
32	trade agent error,please check param	交易代理额度失败
33	trade customers error,please check param	交易玩家额度失败
34	billno Repeat	交易号重复
1000	internal error	内部错误<调用接口异常>
1001	agent is illegal	代理非法
1002	send a request too fast	请求太快
1101	params is illegality	参数非法【检查参数加密 KEY 与代理线密码是否一致】
1102	ip is illegality	不是白名单用户
1103	request timeout	请求超时
11000		转账单号已经存在
21001		创建会员账号时登录名称已经存在
60003		创建会员账号时代理非五级代理或者盘口不存在

3.2 APP 与 MH5 交互接口

注：本接口只适用于产品 APP 调用远程 H5 游戏，需要进行 H5 与原生部分通信时，不能用于手机移动 web 站。

1. 竖版设计分辨率为：750X1137

2. 接口生效传参：

产品 APP 调用远程 H5 游戏时，需要给 GCI 传参：

给 GCI 传参：webApp=true 同 gameType, videoID 类似传参。

不传：则下面通知接口不会发送。

注意：手机移动 web 站调用 H5 时，不能传此参数，请注意区分，否则下面 url 无拦截打开空白。

2．**接口定义**

采用 open 一个指定 url，原生 WebView 进行 OverrideUrl 拦截消费，进行接口通信。

url 接口：

https://localhost/xxx.html

或

http://localhost/xxx.html

xxx.html 的定义如下：

portrait.html	通知 APP 只能竖屏
landscape.html	通知 APP 只能横屏
sensor.html	通知 APP 可横竖屏切换
disconnect.html	通知 APP 已断线，需要重新加载游戏
exit.html	通知 APP 用户退出游戏[20200305 新增]

3.3 APP 内嵌 MH5 传参

产品 APP 接入 MH5，但不需要与 MH5 进行交互，则给 GCI 传参：inApp=true，以表示 MH5 是 APP 内运行。传参方式同 gameType, videoID 类似传参。

注：手机浏览器移动 web 站调用 MH5 时，则不传此参数。

3.4 参数值参考说明

3.4.1 webApp，inApp 参数说明

传参场景如下：

产品 APP 内嵌 MH5：

1. APP 需要与 MH5 交互，则传 webApp，接入详见 3.2 部分。
2. 不需要与 MH5 交互，则传 inApp，见 3.3 部分。

手机浏览器 Web 站：

1. 不传以上 2 个参数。

3.4.2 lang 参数说明

终端类型	支持参数
h5	ZH ： 中文 EN ： 英文 JA ： 日文 VI ： 越南文 TH ： 泰文
pch5	ZH ： 中文 EN ： 英文 JA ： 日文 VI ： 越南文 TH ： 泰文

3.4.3 gameType,videoID 参数说明

- 只传 gameType ： 网站进入指定类型游戏列表
- 传 gameType 与对应类型的 videoID ： 网站进入指定游戏

当前游戏可提供如下游戏类别 ：(供参考，以实际桌台为准)

游戏类别	gameType	videoID	视频名字（游戏可见）
经典百家乐	BAC	B001,	B01

		B002,	B02
		B003,	B03
		B004,	B04
		B005,	B05
		B006,	B06
		C001,	C01
		C002,	C02
		C003,	C03
		C005,	C05,
		C006	C06
		D051	D51
		D052	D52
		D053	D53
		D054	D54
		D055	D55
		D056	D56
		D057	D57
		D058	D58
		D059	D59
		D070	D70
		D071	D71
		D072	D72

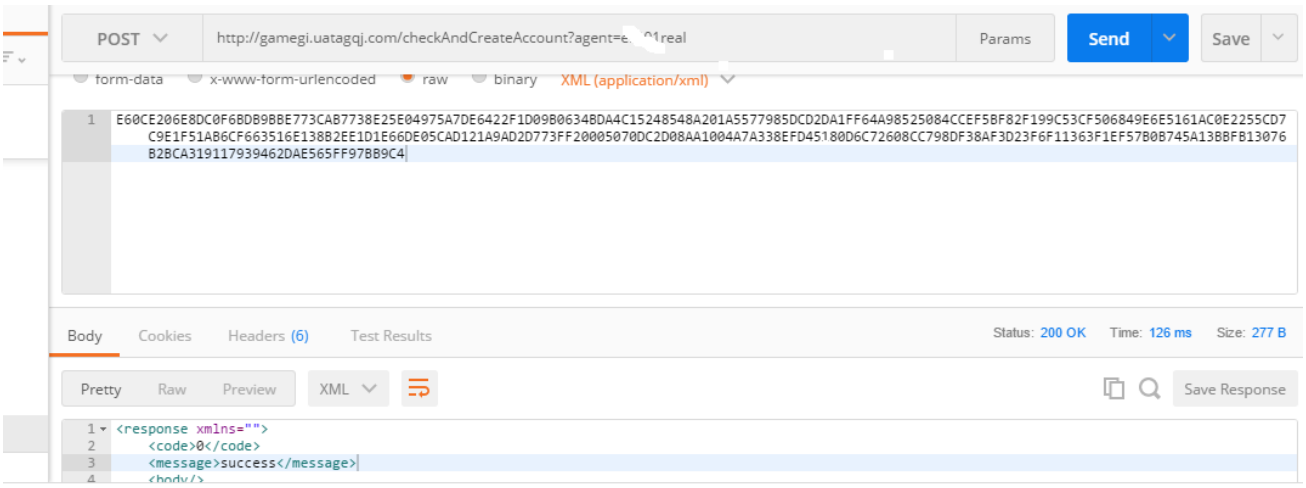
		D073 N001 N002 N003 N004 N005	D73 N01 N02 N03 N04 N05
包桌百家乐	CBAC	T023, T024, T025, T026, T027, T028, T029, T030	T23 T24 T25 T26 T27 T28 T29 T30
保险百家乐	SBAC	B007	B07
连环百家乐	LINK		
龙虎	DT	B021, C007 D075	B21 C07 D75
骰宝	SHB	B025 C015	B25 C15
炸金花	ZJH	C016	C16

轮盘	ROU	B023, C008 D001	B23, C08 D01
牛牛	NN	B022	B22
斗牛	BF	C019	C19
百家乐大赛	RACE		

3.5 加密串校验示例

key明文:xdr56yhn
key MD5密文:5e517f9e96874533149735c18707a102

Params参数明文:{"timestamp":"1111","userName":"mina1234561"}
加密后的密文【不区分大小写】:
7c6f088329563b2979794f4e0769b8bac5fb071ecbadb333f55fe630bbdde6cdcb4f48b0c5fe9211da92dea4a0f483b2



3.6 Java 加密算法示例

```
String key = CipherUtils.string2MD5CipherText(authPwd);
String params = CipherUtils.string2AESCipherText (key, json);
```

```
     /**
     * @param inStr
     * @return String
     * @Description:<字符串加密成md5密文>
     */
    public static String string2MD5CipherText(String inStr) {
        byte[] string2md5Byte = string2MD5CipherByte(inStr);
        return parseByte2HexStr(string2md5Byte);
    }

    /**
     * @param inStr
     * @return String
     * @Description:<字符串加密成md5二进制数组>
     */
    public static byte[] string2MD5CipherByte(String inStr) {
        if (inStr == null) {
            return null;
        }
        MessageDigest md5 = null;
        try {
            md5 = MessageDigest.getInstance(MD5_TYPE);
            byte[] bytes = inStr.getBytes(ENCODING);
            return md5.digest(bytes);
        } catch (Exception e) {
            Logger.error("error ->
string2MD5CipherByte(),param:{},error:{}", inStr, e);
            return null;
        }

    }


    /**
     * @param key  加密的key
     * @param text 需要加密的字符串
     * @return String字符串密文
     * @Description:<字符串加密成AES密文>
     */
    public static String string2AESCipherText(String key, String text) {
        byte[] cipherByte = string2AESCipherByte(key, text);
        return parseByte2HexStr(cipherByte);
    }
/**
```

```
     * @param md5Key 编码后的AESkey
     * @param text      待加密的字符串
     * @return byte[]加密后的byte[] 数组
     * @Description:<根据加密key加密字符串>
     * @Time:2017年12月26日下午1:55:13
     */
    public static byte[] string2AESCipherByte(String md5Key, String text)
{
        byte[] key = parseHexStr2Byte(md5Key);
        SecretKeySpec sKeySpec = new SecretKeySpec(key, AES_TYPE);
        byte[] cipherByte = null;
        try {
            Cipher cipher =
            Cipher.getInstance(AES_TYPE);//AES/ECB/PKCS5Padding
            cipher.init(Cipher.ENCRYPT_MODE, sKeySpec);
            cipherByte = cipher.doFinal(text.getBytes(ENCODING));
        } catch (InvalidKeyException e) {
            Logger.error("error ->
                  string2AESCipherByte(),param:{{},{}},error:{}", md5Key,
                  text, e);
        }
        return cipherByte;
    }
```

3.7 Node JS 加密算法示例

```
// encrypt
const cipher = crypto.createCipher('aes-128-ecb', authPwd);
var crypted = cipher.update(params, 'utf8', 'hex');
crypted += cipher.final('hex');
console.log("参数加密后：" + crypted);
```

3.8 PHP 加密算法示例

```
<?php
class CryptAES
{
    protected $cipher = MCRYPT_RIJNDAEL_128;
    protected $mode = MCRYPT_MODE_ECB;
    protected $pad_method = NULL;
    protected $secret_key = '';
    protected $iv = '';
```

```
public function set_cipher($cipher)
{
	$this->cipher = $cipher;
}

public function set_mode($mode)
{
	$this->mode = $mode;
}

public function set_iv($iv)
{
	$this->iv = $iv;
}

public function set_key($key)
{
	$this->secret_key = $key;
}

public function require_pkcs5()
{
	$this->pad_method = 'pkcs5';
}

protected function pad_or_unpad($str, $ext)
{
	if ( is_null($this->pad_method) )
	{
		return $str;
	}
	else
	{
		$func_name = __CLASS__ . '::' . $this->pad_method . '_' . $ext . 'pad';
		if ( is_callable($func_name) )
		{
			$size = mcrypt_get_block_size($this->cipher, $this->mode);
			return call_user_func($func_name, $str, $size);
		}
	}
	return $str;
}
```

```
protected function pad($str)
{
	return $this->pad_or_unpad($str, '');
}

protected function unpad($str)
{
	return $this->pad_or_unpad($str, 'un');
}

public function encrypt($str)
{
	$str = $this->pad($str);
	$td = mcrypt_module_open($this->cipher, '', $this->mode, '');

	if ( empty($this->iv) )
	{
		$iv = @mcrypt_create_iv(mcrypt_enc_get_iv_size($td), MCRYPT_RAND);
	}
	else
	{
		$iv = $this->iv;
	}

	mcrypt_generic_init($td, hex2bin($this->secret_key), $iv);
	$cyper_text = mcrypt_generic($td, $str);
	$rt = strtoupper(bin2hex($cyper_text));
	mcrypt_generic_deinit($td);
	mcrypt_module_close($td);

	return $rt;
}

public function decrypt($str){
	$td = mcrypt_module_open($this->cipher, '', $this->mode, '');

	if ( empty($this->iv) )
	{
		$iv = @mcrypt_create_iv(mcrypt_enc_get_iv_size($td), MCRYPT_RAND);
	}
	else
	{
		$iv = $this->iv;
	}
```

```
        mcrypt_generic_init($td, $this->secret_key, $iv);
        //$decrypted_text = mdecrypt_generic($td, self::hex2bin($str));
        $decrypted_text = mdecrypt_generic($td, base64_decode($str));
        $rt = $decrypted_text;
        mcrypt_generic_deinit($td);
        mcrypt_module_close($td);

        return $this->unpad($rt);
    }

    public static function hex2bin($hexdata) {
        $bindata = '';
        $length = strlen($hexdata);
        for ($i=0; $i< $length; $i += 2)
        {
            $bindata .= chr(hexdec(substr($hexdata, $i, 2)));
        }
        return $bindata;
    }

    public static function pkcs5_pad($text, $blocksize)
    {
        $pad = $blocksize - (strlen($text) % $blocksize);
        return $text . str_repeat(chr($pad), $pad);
    }

    public static function pkcs5_unpad($text)
    {
        $pad = ord($text{strlen($text) - 1});
        if ($pad > strlen($text)) return false;
        if (strspn($text, chr($pad), strlen($text) - $pad) != $pad) return false;
        return substr($text, 0, -1 * $pad);
    }
}

//密钥
$keyStr =md5('xdr56yhn');
//加密的字符串
$plainText = '{"timestamp":"1111","userName":"mina1234561"}';

$aes = new CryptAES();
$aes->set_key($keyStr);
$aes->require_pkcs5();
```

```
$encText = $aes->encrypt($plainText);

echo $encText;

?>
```

3.9 C++加密算法示例

```
//需要将16进制的md5字符串转成16进制数组
const uint8_t md5key[16] =
{ 0x5e,0x51,0x7f,0x9e,0x96,0x87,0x45,0x33,0x14,0x97,0x35,0xc1,0x87,0x07,0xa1,0x02 };//md5值

void AES_ECB_ENCRYPTSTR(const std::string& src)
{
    cipher::AES_ECB_Cipher  cipher(md5key);  // cipher::AES_ECB_Cipher是开源类，参考下载地址
    uint8_t dest[1024];
    uint32_t destlen = sizeof(dest);
    if (0 != cipher.encode((const uint8_t *)src.c_str(), src.length(), dest, destlen))//开源
方法
    {
        return;
    }
}
```

AES 加密开源方法下载地址:
https://github.com/HUTOYP/AES128_ECB_PKCS5Padding

4 Q&A

1.用户登录 是用来做什么的？ http://GameGI.Domain/userLogin?agent=xxx
该接口返回 ticket 进入游戏调用 gameEntrance 接口时需要此 ticket 参数值，我们好像没发现有 gameEntrance 接口
----用户登录接口 是获取进入游戏的 ticket 验证信息的接口，进游戏时需要把 ticket 作为参数传入，gameEntrance 是提供的 gci_web 系统的接口。

2.验证并创建账号 无法区分是已有的账号还是新的账号，因为返回值都是 0.
注册时，输入已注册过的账号也提示注册成功，应该提示已注册
----这个接口的意义就是，没有账号就创建，有账号就返回成功，调用方在获取返回值为 0 的情况就可以判定这个账号在厅方已经存在。

3.转账是不是我们的代理号给游戏号转账，那我们是不是要给自己的代理号充钱

----转账就是用户在你们网站的额度转入到厅方游戏中，跟代理号没有关联关系

4.会员总额度=网站额度+“getBalance”额度 网站额度指的是什么？
---- 网站额度就是你们调用方用户的自己额度

5. 存入时，预转账返回 code=0，正常，确认转账返回 code 1000,'message' => 'internal error',
---- 预转账只是先检验一下用户的相关信息，确认转账才是真实的把转入的额度加在厅方该用户的额度字段。所有预转账成功，确认转账也有可能失败是正常的【eg:用户在这两个接口调用时间差有账户额度变更】。

6. 取出时，预转账返回'code'= '20','message' => 'invalid credit',是代表玩家金币不足？
--用户的额度字段 `credit` 必须为大于 0 的正整数。

7. APP 内 MH5 游戏运行时断网或其它原因出现 404 空白页面错误？
--请检查 404 的地址是什么？出现此问题，有如下 2 个原因：
--如果有传 webApp=true 参数，但原生 WebView 没有做 OverrideUrl 拦截消费;
--如果没有传 webApp 参数，则必须给 GCI 设置回调地址，断开网络会回调打开网站首页地址;