

九游游戏_SDK_开发参考说明书_服务器接口_v1.2.12API

2019-05-23

修订记录

归档日期	版本	说明	作者	审批人
2012-07-17	1.0.7	增加支付通道代码（1.3.3）		
2012-07-20	1.0.8	补充加密方式和支付通道的说明（1.3.3）		
2012-08-09	1.0.9	增加新支持的支付通道说明（1.3.3）		
2012-08-24	1.0.10	增加游戏老账号登录与九游账号进行绑定的说明（2.游戏服务器开发要点）		
2012-08-27	1.0.11	1.新增角色查询接口说明（1.3.4）； 2.充值结果回调接口增加角色编号字段（1.3.3）。		
2012-11-09	1.0.12	1.充值结果回调接口增加充值场景可选字段，及PC反充说明（1.3.3）； 2.增加新支持的支付通道说明（1.3.3）。		
2013-07-02	1.0.13	更新支付通道代码（1.3.3）		
2014-07-14	1.0.14	更新充值结果回调接口的说明（1.3.3）		
2014-07-23	1.0.15	1.更新用户会话验证的说明（1.3.1） 2.更新充值结果回调接口的说明（1.3.3）		
2014-08-04	1.0.16	1.去除过期接口“角色查询接口” 2.新增“扩展数据提交接口”接口说明		
2014-08-05	1.0.17	修改各接口中channelId的说明		
2014-09-02	1.1.0	1.去掉请求参数game中cpId,serverId,channelId 2.用户会话验证接口修改为account.verifySession		

2014-10-14	1.2.0	签名规则去掉cpId值		
2014-11-21	1.2.1	更新充值结果回调接口的说明 (1.3.3)		
2014-11-21	1.2.2	更新充值结果回调接口的说明 (1.3.3)		
2014-11-24	1.2.3	更新充值结果回调接口的说明 (1.3.3)		
2015-03-18	1.2.4	1.更新协议说明 (1.2.2)		
		2.更新登录会话验证接口说明 (1.3.1)		
2015-06-10	1.2.5	1.更新协议说明 (1.2.2)		
		2.更新充值结果回调接口 (1.3.3)		
		3.删除原“登录状态同步接口” (1.3.2)		
2015-10-27	1.2.6	1.新增关于昵称nickName的使用说明 (1.3.1)		
2016-02-26	1.2.7	1.修改及新增扩展数据接口类型字段 (1.3.3)		
		2.扩展数据提交接口”接口提交类型说明 (1.3.3)		
2016-03-03	1.2.8	1.游戏数据上报接口协议去除sid参数, 只使用accountId参数 (1.3.3)		
2016-10-21	1.2.9	服务端域名由g.uc.cn变更为9game.cn		
2016-11-10	1.2.10	新增角色创建者: WDJ: 豌豆荚、ALI: 阿里游戏		
2017-05-04	1.2.11	新增查询玩家是否身份证实名接口 (1.3.5)		
2019-05-23	1.2.12	身份证实名接口新增返回玩家是否成年字段		

1. 服务器端接口说明

1.1. 概要

本部分主要提供九游游戏中心“SDK服务器”和游戏合作商“游戏服务器”的交互的接口规范。

1.2. 协议说明

1.2.1 通信协议

“SDK服务器”采用HTTP协议作为通信协议，“游戏服务器”通过构造HTTP请求（POST方式）向“SDK服务器”发起接口请求。

“游戏服务器”HTTP请求数据示例：

```
POST http://sdk.9game.cn/cp/account.verifySession HTTP1.1
Content-Type: application/json
Content-Length: 136
{
  "id":1332406591685,
  "game":{"gameId":5},
  "data":{"
    "sid":"110adf4c-f2d3-4be5-8a9c-3741a83e5853"
  },
  "sign":"bb926c2a9944e9b4f2f6639d928dc95c"
}
```

“SDK服务器”响应数据示例：

```
200 OK
Content-Type: application/json;charset=utf-8
Content-Length: 155

{
  "id":1332406591685,
  "state":{"code":1, "msg":"操作已完成"},
  "data":{"
    "creator":"JY",
    "accountId":"U11626774a4e39c16cf7mmsnz5002une",
    "nickName":"昵称"
  }
}
```

1.2.2数据协议

1) 数据格式

请求消息和响应消息的内容都使用json格式表示数据，具体请参考下文的例子。

2) 字符编码

请求与响应内容（json格式数据）须采用utf-8字符编码。

3) 签名规则

使用md5哈希对请求内容进行签名，算法如下：

```
md5(签名内容+apiKey)
```

说明：

- MD5使用RFC 1321标准，编码后需要转换回全小写(本文提到的所有md5结果，都需要转换成全小写)。
- 表达式中的“+”号表示字符串接，并不存在。
- 签名内容是指请求数据（data字段）中，各字段名及其字段值的拼接，字段名与字段值之间使用等号（=）连接。拼接时需对字段名排序，排序方式是按字段名进行升序排列。
- **注意：签名内容不应包含“&”符号，拼接签名内容时需把“&”符号剔除。计算MD5签名时，取签名内容的字节时，应以UTF-8编码取字符串的字节值。**
- apiKey由九游分配，游戏在审核通过后可通过开放平台查看。

4) 签名示例

假设请求内容请求数据（data字段）为：

```
"data":{
  "personid":value1,
  "code":value2",
  "name":value3"
}
```

排序拼接后得出要签名的内容串为：

```
code=value2name=value3personid=value1
```

假定apiKey=202cb962234w4ers2aa,

要进行MD5哈希的字符串为：

```
echo -n code=value2name=value3personid=value1202cb962234w4ers2aa|md5sum
```

执行MD5哈希，下面为在linux shell中执行命令取得MD5值：

```
8f468d03c2c42fbe294bdd49f038c031
```

得出签名结果是：

```
8f468d03c2c42fbe294bdd49f038c031
```

最后发送的请求内容为：

```
{
  "id":1330395827,
  "service":"aa.aa.aa",
  "data":{
    "personid":value1,
    "code":"value2",
    "name":"value3"
  },
  "sign":"8f468d03c2c42fbe294bdd49f038c031"
}
```

1.3. 接口说明

1.3.1 用户会话验证接口（必接）

如游戏在豌豆荚上线过，并且接入九游使用老接口协议ucid.user.sidInfo的，需要更新为本新接口，否则无法返回creator字段标识用户来源,更新接口前需联系技术接口人调整SDK内定义的接口协议类型，否则调用接口会返回“178，调用接口类型错误”

1) 请求地址

<http://sdk.9game.cn/cp/account.verifySession>

2) 调用方式：HTTP POST

3) 接口描述：

验证sid是否为有效的登录用户会话，若有效则返回其账号标识、账号创建者和昵称。

“游戏客户端”通过“SDK客户端”获取到sid（详细参考对应API分册），传到“游戏服务器”，“游戏服务器”到“SDK服务器”验证用户会话sid的有效性，获取用户的账号标识、账号创建者和昵称，供游戏使用。

注意：进行接口调用前请确认sid是否具备值，如sid值为空时请勿调用此接口。

接入有疑问请戳这里：<http://bbs.9game.cn/thread-16364522-1-1.html>

4) 请求方：游戏服务器

5) 响应方：SDK服务器

6) 请求内容（json格式）：

字段名称	字段说明	类型	必填	备注
id	请求的唯一标识	long	Y	Unix时间戳，例： 1330395827
data	请求数据	json	Y	json格式
game	game参数	json	Y	json格式，字段值 均为整数。格式如 下： { “gameId”：游戏编 号 }
sign	签名参数	string	Y	MD5(签名内容 +apiKey) 签名内容： sid=.....
请求数据(对应data,采用json格式)				
sid	当前用户会话标识	string	Y	游戏客户端登录后 从SDK取得

例子：

HTTP请求的body内容：

```
{  
  "id":1330395827,  
  "data":{"sid":"abcdefg123456"},  
  "game":{"gameId":12345},  
  "sign":"6e9c3c1e7d99293dfc0c81442f9a9984"  
}
```

假定apiKey=202cb962234w4ers2aaa

sign的签名规则：MD5(sid=...+apiKey)（去掉+；替换...为实际值）

签名原文：

sid=abcdefg123456202cb962234w4ers2aaa

MD5加密：

echo -n sid=abcdefg123456202cb962234w4ers2aaa|md5sum

加密结果：091391c3613711383d4d631318674ac8

7) 返回内容 (json格式)：

字段名称	字段说明	类型	必填	备注
id	请求的唯一标识	long	Y	Unix时间戳，与请求内容中id值相同。例： 1330395827
state	响应状态	json	Y	
data	响应数据	json	Y	
响应状态（对应state,采用json格式）				
code	响应码	int	Y	
msg	结果描述	string	Y	
响应数据(对应data,采用json格式)				
accountId	账号标识	string	Y	最长为32个字符
creator	账号创建者	string	Y	JY：九游 PP：PP助手 WDJ：豌豆荚 ALI：阿里游戏 全新游戏上线，则统一返回ALI，如游戏在九游上线过，用户登录九游包则返回JY，如游戏在豌豆荚上线过，用户登录豌豆荚包则返回WDJ，如游戏在PP上线过，用户登录PP包则返回PP
nickName	用户昵称	string	Y	长度：20个字符内 字符编码：不支持三位以上的字符编码，不支持表情（含emoji）

				creator：ALI本字段返回空，其它按原返回，建议开发商弱化该字段使用，后续逐渐淘汰
--	--	--	--	--

例子：

```

{
  "id":1330395827,
  "state":{"code":1, "msg":"操作成功"},
  "data":{"
    "accountId":"U11626774a4e39c16cf7mmsnz5002une",
    "creator":"JY",
    "nickName":"九游玩家"
  }}
}

```

响应码说明（state.code）：

响应码	说明	错误原因
1	成功	--
10	请求参数错误	请求内容格式有误、gameID有误、签名校验失败等
11	用户未登录	sid不存在，请求地址有误等
99	服务器内部错误	接口名有误、请求地址有误等

1.3.2充值加签处理（必接）

1) 接口描述：

游戏客户端在调起SDK支付界面前，需从服务器获取订单信息，订单信息内包含游戏服务器需做的加签处理，加签生成签名sign及其它订单信息返回给客户端，客户端再将sign通过支付接口传递给SDK，用于校验订单的准确性，防止用户在下单过程中篡改订单信息，避免造成结算数据不匹配，或经济损失等

2) 请求方：游戏客户端

3) 响应方：游戏服务器

4) 请求内容：游戏自定义，以下为支付所需的SDKParams字段参数说明：

参数定义		类型	是否必填	说明
CALLBACK_INFO	callbackInfo	String	否	

				cp自定义信息，在支付结果通知时回传,CP可以自己定义格式,长度不超过250
AMOUNT	amount	String	是	充值金额，如果传递为0表示用户可以自定义充值金额,最多保留小数点后2位，单位为元。例：10.00，整数金额可不传小数点部分，建议金额由 服务端格式化好，客户端直接使用
NOTIFY_URL	notifyUrl	String	否	服务器通知地址，如果为空则以开放平台配置地址作为通知地址，优先取客户端地址(建议通过 游戏服务端配置参数 后读取)，长度不超过100
CP_ORDER_ID	cpOrderId	String	否	cp充值订单号，需要保证用户每次充值订单号的唯一性，长度不超过30
ACCOUNT_ID	accountId	String	是	服务端调用登录会话验证接口verifySession返回的accountId，即用户唯一标识
SIGN_TYPE	signType	String	是	签名类型，MD5或者RSA 目前只支持MD5
SIGN	sign	String	是	签名结果。 MD5(签名内容+apiKey); 服务端生成 签名算法参考下面说明

注：sign必须由服务器生成，另以上所有参数值请从服务器获取，不能直接在客户端生成，确保客户端与服务器数据一致，并且apikey不允许写在客户端

签名算法：

- (1) 将客户端传递的支付请求信息中所有参数名称key按照字典顺序排列(a-z)，值为空串需参与签名，值为null的参数不需要参与签名，sign和signType也不参数签名。（注意签名的key为上表内第二列“参数名称”，非第一列的“参数定义”）
- (2) 所有参数名称按照参数名的字典排序连接起来组成待签名数据，格式：p1=v1p2=v2p3=v3...，签名内容不应包含“&”符号，拼接签名内容时需把“&”符号剔除
- (3) 根据signType的设置，进行相应MD5或RSA签名；
- (4) 对于MD5签名，将上述方式生成的待签名数据，拼上秘钥后进行MD5运算。
- (5) 对于RSA签名，则是将上述方式生成的待签名数据和秘钥，直接进行RSA签名；
- (6) MD5签名结果忽略大小写；

假如apikey为123456，待签名原始串示例如下（签名内容+apiKey）：

accountId=123452132amount=100.00callbackInfo=xxxxxcpOrderId=XXXXXXnotifyUrl=http://192.168.1.1/notifypage.do123456

签名算法示例代码（参考服务端接入包含集PayChargeSignService实现）：

```
/**
 * 签名工具方法
 * @param reqMap
 * @return
 */
public static String sign(Map<String, String> reqMap, String signKey) {
    //将所有key按照字典顺序排序
    TreeMap<String, String> signMap = new TreeMap<String, String>(reqMap);
    StringBuilder stringBuilder = new StringBuilder(1024);
    for (Map.Entry<String, String> entry : signMap.entrySet()) {
        // sign和signType不参与签名
        if ("sign".equals(entry.getKey()) || "signType".equals(entry.getKey())) {
            continue;
        }
    }
}
```

```

//值为null的参数不参与签名

if (entry.getValue() != null) {

    stringBuilder.append(entry.getKey()).append("=").append(entry.getValue());

}

}

//拼接签名秘钥

stringBuilder.append(signKey);

//剔除参数中含有的'&'符号

String signSrc = stringBuilder.toString().replaceAll("&", "");

return md5(signSrc).toLowerCase();

```

1.3.3充值结果回调接口（如开计费必接）

1) **请求地址：**即充值结果通知地址，由游戏合作商提供。游戏接入时，由游戏合作商可在SDK客户端设置或开放平台录入，优先取客户端设置的地址。

2) **调用方式：**HTTP POST

3) **接口描述：**

用户在游戏中提交充值请求后，九游游戏平台会异步执行充值，在充值操作完成后，九游游戏平台通过该接口将充值结果给“游戏服务器”。

此处定义本接口的规范，游戏合作商需根据此规范在“游戏服务器”实现本接口。

此接口用于接收通过游戏SDK充值的结果通知，和通过PC页面进行直接充值的结果通知。

注：支付回调异步信息是通过把回调信息json串放在http的body里，post给游戏服务器的回调地址上，游戏接收回调时请注意需获取body内容。

接入有疑问请戳这里：<http://bbs.9game.cn/thread-16364522-1-1.html>

4) **请求方：**SDK服务器

5) **响应方：**游戏服务器

6) **请求内容（json格式）：**

字段名称	字段说明	类型	必填	备注
ver	版本号	string	Y	接口版本号，2.0

data	支付结果数据	json	Y	
sign	签名参数	string	Y	MD5(签名内容+apiKey); 签名内容为data所有子字段按字段名升序拼接（剔除&符号及回车和换行符）
支付结果数据(对应data,采用json格式)				
orderId	充值订单号	string	Y	此订单号由九游游戏SDK生成，游戏客户端在进行充值时从SDK获得。
gameId	游戏编号	string	Y	由九游分配
accountId	账号标识	string	Y	游戏服务端必须根据九游回调的账号标识进行 校验用户登录账号与充值账号 一致后再下发相应虚拟货币
creator	账号的创建者	string	Y	JY：九游 PP：PP助手 WDJ：豌豆荚 ALI：阿里游戏
payWay	支付通道代码	string	Y	统一费率“999”，不提供详细充值方式
amount	支付金额	string	Y	单位：元。 游戏服务端必须根据九游回调的金额值进行校验并下发相应价值的虚拟货币。
callbackInfo	游戏合作商自定义参数	string	Y	游戏客户端在充值时传入，SDK服务器不做任何处理，

				在进行充值结果回调时发送给游戏服务器，该字段建议使用字母或数字形式传入，建议不要传入特殊字符。(注意：长度不超过250)
orderStatus	订单状态	string	Y	S-成功支付 F-支付失败，游戏需判断S的订单再下发道具
failedDesc	订单失败原因详细描述	string	Y	如果是成功支付，则为空串。
cpOrderId	Cpi订单号	string	N	仅当客户端调用支付方法传入了cpOrderId参数时，才会将原内容通过透传回游戏服务端。该参数有传递时，才需加入签名，如无，则不需要加入签名(注意：长度不超过30)

请求示例:

```
HTTP请求的body内容:
{
  "ver": "2.0",
  "data":{
    "orderId":"abcf1330",
    "gameId":123,
    "accountId":"12221222211123",
    "creator":"JY",
    "payWay":1,
    "amount":"100.00",
    "callbackInfo":"custominfo=xxxxx#user=xxxx",
    "orderStatus":"S",
    "failedDesc":""
  }
}
```

```
"cpOrderId":"1234567"
},
"sign":"6362e564f832d2e8bbcbd50e75409d47"
}
```

假定apiKey=202cb962234w4ers2aaa

sign的签名规则：

MD5(accountId=...+amount=...+callbackInfo=...+cpOrderId=...+creator=...+failedDesc=...+gameId=...+orderId=...+orderStatus=...+payWay=...+apiKey) (去掉+；替换...为实际值)

注意：cpOrderId仅当回调时具备该参数时，才需加入签名，如无，则不需要加入签名，下同

签名原文：

accountId=12221222211123amount=100.00callbackInfo=custominfo=xxxxx#user=xxxcpOrderId=1234567creator=JYfailedDesc=gameId=123orderId=abcf1330orderStatus=SpayWay=1202cb962234w4ers2aaa

MD5加密：

echo -n

accountId=12221222211123amount=100.00callbackInfo=custominfo=xxxxx#user=xxxcpOrderId=1234567creator=JYfailedDesc=gameId=123orderId=abcf1330orderStatus=SpayWay=1202cb962234w4ers2aaa|md5sum

加密结果：6362e564f832d2e8bbcbd50e75409d47

7) 返回内容（文本）：

响应内容	响应内容描述
SUCCESS	成功，表示游戏服务器成功接收了该次充值结果通知
FAILURE	失败，表示游戏服务器无法接收或识别该次充值结果通知，如：签名检验不正确、游戏服务器接收失败

结果示例：

SUCCESS

8) 通知机制：

充值操作完成后，“SDK服务器”都会将充值结果通过“充值结果回调接口”发送到“游戏服务器”。“游戏服务器”收到“SDK服务器”的充值通知后，根据处理结果返回字符串SUCCESS或FAILURE。如果返回SUCCESS，则“SDK服务器”结束通知任务；如果返回FAILURE或由于网络延时导致“SDK服务器”没有收到任何返回，SDK服务器将会在周期内进行重复通知。

“游戏服务器”在接收“SDK服务器”的充值结果通知时，**返回SUCCESS**，表示不需要“SDK服务器”再次发起通知。当业务逻辑异常（如：收到的SDK服务器的充值结果通知内容的签名不正确、充值结果与提交的充值请求不符等），认为需要再次通知，才返回FAILURE。

【注意】：

由于存在多次发送通知的情况，因此“游戏服务器”必须能够正确处理重复的通知。当接收到通知时，需要检查系统内对应业务数据的状态，以判断该通知是否已经处理过。在对业务数据进行状态检查或处理之前，需要采取数据加锁或时间戳判断等方式进行并发控制。

由于支付网关的通知机制原因，偶尔会发生通知支付失败后又通知支付成功的现象。基于这个情况，“游戏服务器”在处理充值结果通知时，对同一个订单，如果先接收到支付失败，再接收到支付成功的通知，应以成功的支付结果为准，替换原接收到的失败的支付结果。一旦通知支付成功，不会再发生通知支付失败的情形。

1.3.4 扩展数据提交接口（选接）

1) 请求地址：

<http://collect.sdkyy.9game.cn/ng/cpserver/gamedata/ucid.game.gameData>

2) 调用方式：HTTP POST

3) 接口描述：收集玩家游戏的数据

4) 请求方：游戏服务器

5) 响应方：SDK服务器

6) 请求内容（json格式）：

字段名称	字段说明	类型	必填	备注
id	请求的唯一标识	long	Y	Unix时间戳，例： 1330395827
service	接口服务名称	string	Y	ucid.game. gameData
data	请求数据	json	Y	json格式
game	game参数	json	Y	json格式，字段值 均为整数。格式如 下： { “gameId”：游戏编 号 }
sign	签名参数	string	Y	

				MD5(签名内容+apiKey) 签名内容: accountId=... +gameData=...
请求数据(对应data,采用json格式)				
accountId	账号ID	string	Y	九游平台返回给游戏用于标识用户身份的ID。
gameData	游戏数据	string	Y	1.gameData原本是json对象 2.gameData经过json encode 成为一个字符串 3.该字符串再经过urlencode 请看示例
	category	游戏数据类型	string	Y
	content	游戏数据内容	json	Y 如果玩家数据存在换行符, 则要替换成空串

游戏数据类型（category）为loginGameRole。content中只有一条记录，各字段说明如下：

字段名	字段类型	是否为空	字段描述	提交时机
zoneId	string	必填	区服ID,一个服务器对应一个ID,长度不超过50	1、用户登录游戏成功后创建角色或使用已存在的角色进入游戏后调用 2、当用户的角色等级发生变化后调用 (注：必填参数必须传，可选参数可不传或传-1)
zoneName	string	必填	区服名称，要求与游戏界面展示的服务器名保持一致,长度不超过50	
roleId	string	必填	角色ID，一个角色同一个服ID保持唯一,长度不超过50	

roleName	string	必填	角色昵称,长度不超过50
roleCTime	long	必填	角色创建时间(单位: 秒), 获取服务器存储的角色创建时间, 不可用本地手机时间, 同一角色创建时间不可变, 长度10
roleLevel	string	必填	角色等级, 如游戏存在转生, 转职等, 等级需累加, 长度不超过10
os	string	可选	游戏平台 android或者ios, 小写字母。 默认为android, 不可传其它参数
roleLevelMTime	long	必选	角色等级变化时间(单位: 秒), 必须传服务器时间,长度10位, 等级未发生变化时可传-1或不传该参数

接入有疑问请戳这里: <http://bbs.9game.cn/thread-5370208-1-4.html>

请求示例:

HTTP请求的body内容:

```
{
  "id":1330395827,
  "service":"ucid.game.gameData",
  "data":{"accountId":"c6c2f25f2c56cd5dacf77065f3f4ec0d",
    "gameData":"%7b%22category%22%3a%22loginGameRole%22%2c%22content%22%3a%7b%22roleLevel%22%3a%2288%22%2c%22roleName%22%3a%22+%e8%af%b7%e2%88%9d+%e5%86%8d%e7%bb%99%e6%88%91%e4%b8%80%e6%94%af%e7%83%9f%22%2c%22zoneName%22%3a%22%e7%bb%88%e5%8d%97%e5%b1%b1%e4%b8%8b-%e5%85%b5%e4%b8%b4%e5%9f%8e%e4%b8%8b%22%2c%22roleId%22%3a%2253568193%22%2c%22zoneId%22%3a%22705%2c%22roleCTime%22%3a%221353271378%2c%22os%22%3a%22android%22%2c%22roleLevelMTime%22%3a%221456380919%7d%7d"%7d},
    "game":{"gameId":12345},
    "sign":"6b3d0561f5fecb46e35e35c233be42af"
  }
}
```

假如gameData的数据为:

```
{ "category": "loginGameRole", "content": { "roleLevel": "88", "roleName": " 请∞ 再给我一支烟", "zoneName": "终南山下-兵临城下", "roleId": "53568193", "zoneId": "2705", "roleCTime": "1353271378", "os": "android", "roleLevelMTime": "1456380919" }}
```

那么经过UrlEncode后的字符串为 (注: 该字符串也是作为计算MD5签名的原文内容)

```
%7b%22category%22%3a%22loginGameRole%22%2c%22content%22%3a%7b%22roleLevel%22%3a%2288%22%2c%22roleName%22%3a%22+%e8%af%b7%e2%88%9d+%e5%86%8d%e7%bb%99%e6%88%91%e4%b8%80%e6%94%af%e7%83%9f%22%2c%22zoneName%22%3a%22%e7%bb%88%e5%8d%97%e5%b1%b1%e4%b8%8b-%e5%85%b5%e4%b8%b4%e5%9f%8e%e4%b8%8b%22%2c%22roleId%22%3a%2253568193%22%2c%22zoneId%22%3a%222705%2c%22roleCTime%22%3a%221353271378%2c%22os%22%3a%22android%22%2c%22roleLevelMTime%22%3a%221456380919%7d%7d
```

7) 返回内容 (json格式) :

字段名称	字段说明	类型	必填	备注
id	请求的唯一标识	long	Y	Unix时间戳, 与请求内容中id值相同。例: 1330395827
state	响应状态	json	Y	
data	响应数据	json	Y	
响应状态 (对应state,采用json格式)				
code	响应码	int	Y	
msg	结果描述	string	Y	
响应数据(对应data,采用json格式)				

例子:

```
{
  "id": 1330395827,
  "state": { "code": 1, "msg": "操作已完成" },
  "data": {}
}
```

响应码说明 (state.code) :

响应码	说明
1	成功
10	请求参数错误/无效的请求数据,校验签名失败
11	用户未登录(或无效的sid)

2. 游戏服务器开发要点

“游戏服务器”相关接入开发工作要点如下：

以下3个接口均为必接接口

1.实现sid验证功能：当用户使用“九游账号”登录模式时，接收“游戏客户端”提交的sid，通过调用“SDK服务器”的“用户会话验证”接口，验证sid的合法性，根据需要获取登录用户的账号信息。

2.实现“充值加签功能”：当用户在调起支付界面前，“游戏客户端”提交订单信息给“游戏服务器”做加签处理，返回sign给“游戏客户端”，以保证CP与SDK之间的订单的一致性

3.实现“充值结果回调接口”（包含充值功能时需要实现）：由于充值方式会不断发展，产生新的支付通道，实现“充值结果回调接口”时，应能支持未来新扩展的支付通道。简单地，不应该对接收到的充值结果的支付通道进行限制，只需对用户、游戏、服务器信息时行校验，校验通过则接受收到的充值结果。