

# 九游\_单机游戏支付\_开发参考说明书\_服务器接口\_v1.0.1

2015-07-16

## 修订记录

归档日期	版本	说明	作者	审批人
2015-05-27	1.0.0	初稿	吴亮	
2015-07-16	1.0.1	增加返回支付方式编码		
2017-04-27	1.0.2	CP接入单机SDK服务端接口		

## 1. 服务器接口说明

### 1.1. 概要

本部分主要提供阿里游戏中心“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：

▼ Plain Text 复制代码

```
1 POST http://sgsdkaccount.9game.cn/sg/cp/sgaccount.verifySession HTTP1.1
2 Content-Type: application/json
3 Content-Length: 136
4 {
5     "id":1332406591685,
6     "game":{"gameId":5},
7     "data":{"
8         "sid":"110adf4c-f2d3-4be5-8a9c-3741a83e5853"
9     },
10    "sign":"bb926c2a9944e9b4f2f6639d928dc95c"
11 }
```

“SDK服务器”响应数据示例：

▼ Plain Text 复制代码

```
1 200 OK
2 Content-Type: application/json;charset=utf-8
3 Content-Length: 155
4
5 {
6     "id":1332406591685,
7     "state":{"code":1, "msg":"操作已完成"},
8     "data":{"
9 "creator":"ALI",
10    "accountId":"U11626774a4e39c16cf7mmsnz5002une"
11    }
12 }
```

## 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：

▼ Plain Text 复制代码

```
1 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由阿里分配，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：

▼ Plain Text 复制代码

```
1 "data":{
2     "personid":value1,
3     "code":"value2",
4     "name":"value3"
5 }
6 }
```

排序拼接后得出要签名的内容串为：

▼ Plain Text 复制代码

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

假定apiKey=202cb962234w4ers2aa,

要进行MD5哈希的字符串为：

▼ Plain Text 复制代码

```
1 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：

▼ Plain Text 复制代码

```
1 8f468d03c2c42fbe294bdd49f038c031
```

得出签名结果是：

▼ Plain Text 复制代码

```
1 8f468d03c2c42fbe294bdd49f038c031
```

最后发送的请求内容为：

```
1  {
2      "id":1330395827,
3      "game":{"gameId":5},
4      "data":{"
5          "personid":value1,
6          "code":"value2",
7          "name":"value3"
8      },
9      "sign":"8f468d03c2c42fbe294bdd49f038c031"
10 }
```

## 1.3. 接口说明

### 1.3.1 用户会话验证接口（选接）

#### 1) 请求地址

正式环境：<http://sgsdkaccount.9game.cn/sg/cp/sgaccount.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。

“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：

▼ Plain Text 复制代码

```
1 HTTP请求的body内容:
2 {
3   "id":1330395827,
4   "data":{"sid":"abcdefg123456"},
5   "game":{"gameId":12345},
6   "sign":"6e9c3c1e7d99293dfc0c81442f9a9984"
7 }
8 假定apiKey=202cb962234w4ers2aaa
9
10 sign的签名规则: MD5(sid=...+apiKey) (去掉+; 替换...为实际值)
11
12 签名原文:
13 sid=abcdefg123456202cb962234w4ers2aaa
14
15 MD5加密:
16 echo -n sid=abcdefg123456202cb962234w4ers2aaa|md5sum
17
18 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 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

例子：

▼ Plain Text 复制代码

```
1  {
2      "id":1330395827,
3      "state":{"code":1, "msg":"操作成功"},
4      "data":{"
5  "accountId":"U11626774a4e39c16cf7mmsnz5002une",
6  "creator":"JY"
7  }
8  }
```

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 充值结果通知接口（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，由游戏合作商提供给UC游戏运营人员，录入到UC的游戏平台中。

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，UC游戏平台会异步执行充值，在充值操作完成后，UC游戏平台通过该接口将充值结果发送给“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。

4) 请求方：单机游戏支付服务器

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， 1.0
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)				
tradeId	交易流水号	string	Y	此订单号由UC游戏SDK生成，游戏客户端在进行充值时从SDK获得。
tradeTime	交易时间	String	Y	订单交易时间，格式： yyyyMMddHHmmss，如 20150527130000
orderId	CP充值订单号	String	N	由游戏传递的订单号

gameId	游戏编号	string	Y	由UC分配
amount	支付金额	string	Y	单位：元。
payType	支付方式代码	string	Y	支付通道代码见下文表格(注意该值可能会增加)
attachInfo	游戏合作商自定义参数	string	N	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。
orderStatus	订单状态	string	Y	S-成功支付 F-支付失败
failedDesc	订单失败原因详细描述	string	Y	如果是成功支付，则为空串。

请求示例:

Plain Text | 复制代码

```
1 HTTP请求的body内容:
2 {
3   "ver": "1.0",
4   "data": {
5     "tradeId": "abcf1330",
6     "tradeTime": "20150527130000",
7     "orderId": "1234567",
8     "gameId": 123,
9     "amount": "100.00",
10    "payType": "999",
11    "attachInfo": "custominfo=xxxxx#user=xxxx",
12    "orderStatus": "S",
13    "failedDesc": ""
14  },
15  "sign": " e933f38cd7697605a70c962af6f6a45b"
16 }
17
18 假定 通知签名秘钥=202cb962234w4ers2aaa
19
20 sign的签名规则:
21 MD5(amount=...+attachInfo=...+failedDesc=...+gameId=...
22 +orderId=...+orderStatus=...+payType=...+tradeId=...tradeTime=...+通知签名
  秘钥 (去掉+; 替换...为实际值)
23 注意: 若参数是非必填, 如orderId, attachInfo等参数, 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
24
25 签名原文:
26 amount=100.00attachInfo=custominfo=xxxxx#user=xxxxfailedDesc=gameId=123ord
  erId=1234567orderStatus=SpayType=999tradeId=abcf1330tradeTime=201505271300
  00202cb962234w4ers2aaa
27
28 MD5加密:
29 echo -n
30 amount=100.00attachInfo=custominfo=xxxxx#user=xxxxfailedDesc=gameId=123ord
  erId=1234567orderStatus=SpayType=999tradeId=abcf1330tradeTime=201505271300
  00202cb962234w4ers2aaa|md5sum
31
32 加密结果: e933f38cd7697605a70c962af6f6a45b
```

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，对于充值结果为失败的，只要能成功接收，也应返回SUCCESS。
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：

▼ Plain Text   复制代码	
1	SUCCESS

8) 通知机制：

充值操作完成后，不论是否充值成功，“单机游戏支付服务器”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“单机游戏支付服务器”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“单机游戏支付服务器”结束通知任务；如果返回FAILURE或由于网络延时导致“单机游戏支付服务器”没有收到任何返回，单机游戏支付服务器将会在周期内进行重复通知。

“游戏服务器”在接收“单机游戏支付服务器”的充值结果通知时，不管订单是否成功，只要业务逻辑正常，都应该返回SUCCESS，表示不需要“单机游戏支付服务器”再次发起通知。当业

务逻辑异常（如：收到的单机游戏支付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。

**支付方式代码：**

代码	中文说明	备注
999	UC支付	
305	移动短信	
306	联通短信	
307	电信短信	