UC 优视游戏 SDK

开发参考说明书

— 服务器接口

2012-04-11

手机上网用UC，就是比你快

修订记录

归档日期	版本	说明	作者	审批人
2012-04-11	1.0.0	初稿		
2012-04-20	1.0.1	更正“1.3.2登录状态同步接口”请求示例中的service 名称错误		
2012-05-08	1.0.2	添加通信协议中构造签名内容对"&"符号的说明； 明确充值结果回调接口返回值的说明； 补充充值结果回调充值通道说明。		
2012-05-11	1.0.3	补充充值结果回调充值通道注意事项：游戏服务器应能支持扩展的支付通道。		
2012-05-17	1.0.4	修改充值结果回调接口与实际不一致的数据类型说明		
2012-06-05	1.0.5	补充支付结果通知说明。		
2012-06-08	1.0.6	修改接口中 channelId 参数说明；修改充值结果回调接口示例中有误的参数数据类型。		
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 UC 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）		

目录

1. 服务器端接口说明

1.1. 概要

本部分主要提供 UC 游戏中心“SDK 服务器”和 UC 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.g.uc.cn/ss HTTP1.1
Content-Type: application/json

{
    "service":"ucid.user.sidInfo",
    "id":1332406591685,
    "game":{"cpId":30,"gameId":5,"channelId":"2","serverId":0},
    "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":{
        "ucid":56920,
        "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(cpId+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。字符串中若有换行情况，也应把换行符（回车或换行）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。

cpId 与 apiKey 由 UC 分配，UC 负责游戏接入的人员在游戏接入前会把需要由 UC 分配的数据包括 cpId 和 apiKey 通知给游戏合作商。

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 cpId=109，apiKey=202cb962234w4ers2aa，

要进行 MD5 哈希的字符串为：

```
109code=value2name=value3personid=value1202cb962234w4ers2aa
```

执行 MD5 哈希，下面为在 linux shell 中执行命令取得 MD5 值：

```
echo -n
109code=value2name=value3personid=value1202cb962234w4ers2aa|md5sum
```

得出签名结果是：

```
ee10f266b72f7add68134bfbcc04fe2c
```

最后发送的请求内容为：

```
{
    "id":1330395827,
    "service":"aa.aa.aa",
    "data":{
        "personid":value1,
        "code":"value2",
        "name":"value3"
    },
    "sign":"ee10f266b72f7add68134bfbcc04fe2c"
}
```

1.3. 接口说明

1.3.1 用户会话验证

1) 请求地址：http://sdk.g.uc.cn/ss/

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 ucid 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 ucid 和昵称，供游戏使用。

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.user.sidInfo
data	请求数据	json	Y	json 格式

game	game 参数	json	Y	json 格式，各字段值均为整数。格式如下：
				{
				cpId：游戏合作商编号
				gameId：游戏编号
				channelId：渠道编号
				serverId：服务器编号，如果没有请填写 0。
				}

sign	签名参数	string	Y	MD5(cpId+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,
    "service":"ucid.user.sidInfo",
    "data":{"sid":"abcdefg123456"},
    "game":{"cpId":100,"gameId":12345,"channelId":"2","serverId":0}
,
    "sign":"6e9c3c1e7d99293dfc0c81442f9a9984"
}
```

假定 cpId=109，apiKey=202cb962234w4ers2aaa

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

签名原文：

```
109sid=abcdefg123456202cb962234w4ers2aaa
```

MD5 加密：

```
echo -n 109sid=abcdefg123456202cb962234w4ers2aaa|md5sum
```

加密结果：6e9c3c1e7d99293dfc0c81442f9a9984

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)				

ucid	UC 账号	int	Y	
nickName	用户昵称	string	Y	

例子：

```
{
    "id":1330395827,
    "state":{"code":1, "msg":"会话刷新成功"},
    "data":{"ucid":123456, "nickName":"张三"}
}
```

响应码说明（state.code）：

响应码	说明
1	成功
10	请求参数错误
11	用户未登录

1.3.2 登录状态同步接口

1）请求地址：http://sdk.g.uc.cn/ss/

2）调用方式：HTTP POST

3）接口描述：

用户采用“游戏老账号（游戏官方账号）”方式登录过程，“游戏服务器”完成账号登录验证后，调用本接口，通知“SDK 服务器”同步账号的登录状态。

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.bind.create
data	请求数据	json	Y	json 格式

game	game 参数	json	Y	json 格式，各字段值均为整数。格式如下：
				{
				cpId：游戏合作商编号
				gameId：游戏编号
				channelId：渠道编号
				serverId：服务器编号，如果没有请填写 0。
				}

sign	签名参数	string	Y	MD5(cpId+签名内容+apiKey) 签名内容： gameUser=...
请求数据(对应 data,采用 json 格式)				
gameUser	游戏老账号	string	Y	用户的游戏老账号（游戏官方账号）

例子：

HTTP 请求的 body 内容：

```
{
  "id":1330395827,
  "service":"ucid.bind.create",
  "data":{"gameUser":"abcdefg123456"},
  "game":{"cpId":100,"gameId":12345,"channelId":"2","serverId":10
5},
  "sign":"6e9579ddbaa32aa77b6356df71ffc0e8"
}
```

假定 cpId=109，apiKey=202cb962234w4ers2aaa

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

签名原文：

109gameUser=abcdefg123456202cb962234w4ers2aaa

MD5 加密：

```
echo -n 109gameUser=abcdefg123456202cb962234w4ers2aaa|md5sum
```

加密结果：4a2f6d4459fc905f30dd1e0e7ca12a59

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)				
ucid	对应的 UC 账号	int	Y	

sid	登录会话 id	string	Y	

例子：

```
{
    "id":1330395827,
    "state":{"code":1,"msg":"操作成功"},
    "data":{
        "ucid":123456,
        "sid":"110adf4c-f2d3-4be5-8a9c-3741a83e5853"
    }
}
```

响应码说明：

响应码	说明
1	成功
其它	失败

【注意】：“游戏服务器”在成功调用该接口后，从“SDK 服务器”获得了游戏老账号（游戏官方账号）对应的 UC 账号，必须进行映射绑定，以能够将使用这两个账号登录的用户识别为同一个用户。

1.3.3 充值结果回调接口

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， 和通过 PC 页面进行直接充值的结果通知。

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
data	支付结果数据	json	Y	
sign	签名参数	string	Y	MD5(cpId+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 UC 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 UC 分配
serverId	服务器编号	string	Y	由 UC 分配
ucid	UC 账号	string	Y	
payWay	支付通道代码	string	Y	支付通道代码见下文表格
amount	支付金额	string	Y	单位：元。
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。
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。
roleId	角色编号	string	N	对于通过 PC 页面直接充值的，此参数的值为之充值的角色 ID；对于在 SDK 中充值的，无此参数。
intfType	订单场景	int	N	0 或无此字段：SDK 方式 1：WAP 方式 2：WEB 方式 仅接入了**反充**的游戏有此字段，如游戏同时使用反充和 sdk 支付，需兼容此字段有值和无值的情形。

【**注**】反充：是指在 PC 上进入 UC 的游戏反充页面，用户登录后选择游戏、服务器（分区）、角色后，直接对游戏进行充值，而不需在游戏中进行充值。主要是为了利用 PC 上进行支付的便利，方便玩家在无法使用手机进行充值的情况下可以在 PC 上进行充值。同时适用于客户端游戏和页面游戏。

关于反充，请参考游戏充值 PC 反充的说明文档。

请求示例:

HTTP 请求的 body 内容：

```
{
    "data":{
        "orderId":"abcf1330",
        "gameId":123,
        "serverId":654,
        "ucid":123456,
        "payWay":1,
        "amount":"100.00",
        "callbackInfo":"custominfo=xxxxx#user=xxxx",
        "orderStatus":"S",
        "failedDesc":""
    },
    "sign":"e49bd00c3cf0744c7049e73e16ae8acd"
}
```

假定 cpId=109，apiKey=202cb962234w4ers2aaa

sign 的签名规则：

MD5(cpId + amount=...+callbackInfo=...+failedDesc=...+gameId=...+orderId=...+orderStatus=...+payWay=...+serverId=...+ucid=...+apiKey)（去掉+；替换...为实际值）

签名原文：

```
109amount=100.00callbackInfo=custominfo=xxxxx#user=xxxxfailedDesc=gameId=123orderId=abcf1330orderStatus=SpayWay=1serverId=654ucid=123456202cb962234w4ers2aaa
```

MD5 加密：

```
echo -n 109amount=100.00callbackInfo=custominfo=xxxxx#user=xxxxfailedDesc=gameId=123orderId=abcf1330orderStatus=SpayWay=1serverId=654ucid=123456202cb962234w4ers2aaa|md5sum
```

加密结果：e49bd00c3cf0744c7049e73e16ae8acd

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：

```
SUCCESS
```

8) 支付通道代码：

代码	中文说明	类型	备注
101	U 点 WAP 支付	U 点	
102	U 点直接扣费	U 点	
201	支付宝主账号	支付宝	
202	支付宝安全支付	支付宝	
203	财付通主账户	财付通	
301	移动充值卡	手机充值卡	
302	联通充值卡	手机充值卡	
303	电信充值卡	手机充值卡	
304	移动话费支付	移动话费	
401	骏网一卡通	游戏充值卡	
402	盛大卡	游戏充值卡	
403	征途卡	游戏充值卡	
404	完美一卡通	游戏充值卡	
521	工商银行	PC 网银	
522	建设银行	PC 网银	
523	招商银行	PC 网银	
524	农业银行	PC 网银	
525	中国银行	PC 网银	
526	交通银行	PC 网银	

527	光大银行	PC 网银	
528	民生银行	PC 网银	
529	兴业银行	PC 网银	
530	上海浦东发展银行	PC 网银	
531	广发银行	PC 网银	
532	中信银行	PC 网银	
533	邮政银行	PC 网银	
534	深圳发展银行	PC 网银	
535	平安银行	PC 网银	
536	渤海银行	PC 网银	
537	北京银行	PC 网银	
538	北京农村商业银行	PC 网银	
539	南京银行	PC 网银	
540	上海银行	PC 网银	
541	宁波银行	PC 网银	
542	东亚银行	PC 网银	
543	浙商银行	PC 网银	
600	信用卡-通用快捷支付	手机网银	渠道商为支付宝
601	工商银行信用卡	手机网银	渠道商为支付宝
602	建设银行信用卡	手机网银	渠道商为支付宝
603	招商银行信用卡	手机网银	渠道商为支付宝
611	广发银行信用卡	手机网银	渠道商为支付宝
700	储蓄卡-通用快捷支付	手机网银	渠道商为支付宝
701	工商银行储蓄卡	手机网银	渠道商为支付宝
702	建设银行储蓄卡	手机网银	渠道商为支付宝
703	招商银行储蓄卡	手机网银	渠道商为支付宝
704	农业银行储蓄卡	手机网银	渠道商为支付宝
713	邮政储蓄银行储蓄卡	手机网银	渠道商为支付宝

【**注意**】支付通道会随着业务的发展而扩展，游戏服务器在实现充值结果回调接口时，

不应局限于上表中的支付通道，需要能够接收新出现的支付通道。如果出现未列在上表中的支付通道时，要求游戏服务器仍然能够正确接收充值结果。

上表中会有多个支付通道代码对应同一种支付方式的情况，这是由于为满足不同用户的操作体验而采取了不同的支付形式的结果，不同形式的支付效果是相同的。

9) 通知机制：

充值操作完成后，不论是否充值成功，“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。

建议：

“游戏服务器”在接收“SDK 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，一般都应该返回 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 角色查询接口

1）请求地址：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 GET

3）接口描述：

游戏支持 PC 页面充值时，需要为充值用户提供其角色信息进行选择，由游戏合作商提供该查询接口，让 UC 游戏充值中心获取后为用户展示。

4）请求方：UC 游戏充值中心

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
id	请求的唯一标识	long	Y	Unix 时间戳，例：1330395827
data	角色查询请求数据	json	Y	
sign	签名参数	string	Y	MD5(cpId+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)				
gameId	游戏编号	string	Y	
serverId	服务器分区编号	string	Y	
ucid	UC 账号	int	Y	

例子：

```
HTTP 请求的 body 内容：
{
    "id":1330395827
    "data":{
        "gameId":123,
        "serverId":654,
        "ucid":123456
    },
    "sign":"e49bd00c3cf0744c7049e73e16ae8acd"
}

假定 cpId=109，apiKey=202cb962234w4ers2aaa

sign 的签名规则：
MD5(cpId + gameId=...+serverId=...+ucid=...+apiKey)（去掉+；替换...为实际值）

签名原文：
109amount=100.00callbackInfo=custominfo=xxxxx#user=xxxxfailedDesc=gameId=123orderId=abcf1330orderStatus=SpayWay=1serverId=654ucid=123456202cb962234w4ers2aaa

MD5 加密：
echo -n 109gameId=123serverId=654ucid=123456202cb962234w4ers2aaa|md5sum

加密结果：1f240896278ff117364a1d0fb49ac8aa
```

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)				
rolelist	对应 UC 帐号的游戏角色列表	object	Y	"rolelist":[{"roleid":"12","rolename":" 傲 天 "}, {"roleid":"13","rolename":" 傲 地 "}]

例子：

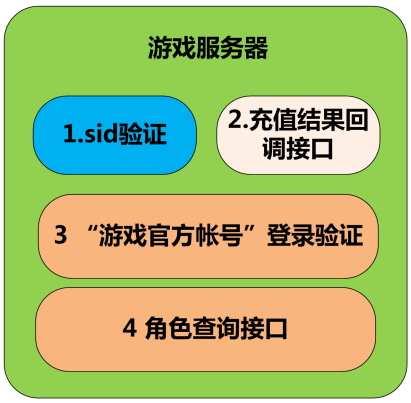
```
{
    "id":1330395827,
    "state":{"code":1,"msg":"操作成功"},
    "data":{
        "rolelist":[
            {"roleid":"12","rolename":"傲天"},
            {"roleid":"13","rolename":"傲地"}
        ]
    }
}
```

响应码说明：

响应码	说明
1	成功
其它	失败

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：



1.实现 sid 验证功能：当用户使用“UC 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 UCID 信息。

2.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。

3.实现“游戏老账号”登录验证（当游戏需要支持“游戏老账号”登录模式时）：

当用户使用“游戏老账号”登录模式时：

a) “游戏服务器”接收“游戏客户端”提交的游戏老账号、密码，进行登录验证；

b) “游戏服务器”调用“SDK 服务器”的“登录状态同步接口”同步用户登录信息，并获取用户对应的 ucid、sid、用户昵称；

c) “游戏服务器”将获得的 ucid 与游戏老账号进行映射绑定（用户下次使用相应的 ucid 登录时，游戏必须将其识别为同一个用户）；

d) “游戏服务器”返回登录验证结果和 sid 信息给“游戏客户端”。

4.实现“角色查询接口”：根据指定的 gameid、serverid 和 ucid，返回该账号在指定游戏服务器下的角色信息列表。