

UC 优视游戏 SDK 开发参考说明书

— 服务器扩展接口

2013-02-20



修订记录

| 归档日期 | 版本 | 说明 | 作者 | 审批人 |
|------------|-------|----|----|-----|
| 2013-02-20 | 1.0.0 | 初稿 | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |



目录

| | |
|-----------------------|----|
| 1. 服务器端增强接口说明..... | 3 |
| 1.1. 概要..... | 3 |
| 1.2. 协议说明..... | 3 |
| 1.2.1 通信协议..... | 3 |
| 1.2.2 数据协议..... | 4 |
| 1.3. 接口说明..... | 6 |
| 1.3.1 获取玩家的密保手机..... | 6 |
| 1.3.2 查询 UC 会员身份..... | 9 |
| 1.3.3 查询 UC 会员信息..... | 12 |



1. 服务器端增强接口说明

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**增强**接口。这些接口不具通用性，仅适用于部分有这方面业务合作关系的游戏合作商。

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 获取玩家的密保手机

1) 请求地址：<http://sdk.g.uc.cn/ss/>

2) 调用方式：HTTP POST

3) 接口描述：

通过登录会话 sid 查询对应的 UC 账号以及绑定手机号码。

“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 和绑定手机号码，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.getMobileBySid |
| 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.getMobileBySid",
  "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 | |
| mobile | 手机号码 | string | Y | 绑定 uc 帐号的手机号, 如果查询不到或未绑定, 则为空 |

例子：

```
{
  "id":1330395827,
  "state":{"code":1, "msg":"操作已完成"},
  "data":{"ucid":123456, "mobile":"13800138000"}
}
```

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 查询 UC 会员身份

1) 请求地址：<http://sdk.g.uc.cn/ss/>

2) 调用方式：HTTP POST

3) 接口描述：

通过登录会话 sid 查询对应的 UC 账号是否为 UC 会员。

“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, 获取用户是否为 UC 会员。

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.vip.isVip |
| 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: <pre>{ 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。 }</pre> |
| 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.vip.isVip",
  "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 | 对应客户端请求的唯一标识 | long | Y | Unix 时间戳，例： 1330395827 |
| state | 响应状态 | json | Y | |
| data | 响应内容 | json | Y | |
| 响应状态 (对于 state,采用 JSON 格式) | | | | |
| code | 响应码 | int | Y | |
| msg | 结果描述 | string | Y | |



响应内容(对应 data,采用 JSON 格式)

| | | | | |
|--------|-------|-----|---|-----------------|
| status | 是否为会员 | int | Y | 0 表示非会员, 1 表示会员 |
|--------|-------|-----|---|-----------------|

例子：

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

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(或无效的 sid) |

1.3.3 查询 UC 会员信息

1) 请求地址：<http://sdk.g.uc.cn/ss/>

2) 调用方式：HTTP POST

3) 接口描述：

通过登录会话 sid 查询对应的 UC 账号会员信息。

“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 UC 账号会员信息。

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.vip.info |
| 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.vip.info",
  "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)

| cmbInfo | | 等级数据 | json | N | 一个 json 对象，构造如下： |
|---------|-----------|-------|--------|---|-------------------------|
| | grade | 等级 | int | N | 1 表示一级会员，2 表示二级 |
| | validFrom | 会员有效期 | string | N | 格式为：yyyy-MM-dd HH:mm:ss |
| | validTo | 会员有效期 | string | N | 格式为：yyyy-MM-dd HH:mm:ss |
| privs | | 特权数据 | json | N | 一个 json 数组对象，构造如下： |
| | enjoy | 是否能享受 | int | N | 1 代表能享受；0 代表不能享受 |
| | pId | 特权 id | string | N | |
| status | | 会员状态 | int | N | 0-非会员 1-有效会员 2-过期会员 |

例子：

```
{
  "id":1330395827,
  "state":{"code":1, "msg":"操作已完成"},
  "data":{
    "cmbInfo":{"grade":1,"validFrom":"2013-01-05 10:38:11","validTo":"2013-06-09 23:59:59"},
    "privs":[
      {"enjoy":0,"pId":"314214"},
      {"enjoy":0,"pId":"849102"},
      {"enjoy":0,"pId":"870402"},
      {"enjoy":0,"pId":"739012"},
      {"enjoy":0,"pId":"098731"},
      {"enjoy":0,"pId":"683062"},
      {"enjoy":0,"pId":"527392"},
      {"enjoy":0,"pId":"468391"},
      {"enjoy":0,"pId":"394912"},
      {"enjoy":0,"pId":"536835"},
      {"enjoy":0,"pId":"120581"},
      {"enjoy":0,"pId":"934180"}
    ],
    "status":1
  }
}
```

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(或无效的 sid) |

