**海外服务端接口文档（韩国）**

**2018-09-04**

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **归档日期** | **版本** | **说明** | **作者** | **审批人** |
| 2018-09-04 | 1.0.0 | 初始 |  |  |
| 2019-02-16 | 1.0.1 | 支付回调请求头增加支付方式字段 |  |  |
| 2020-02-15 | 1.0.2 | 支付回调增加字段 |  |  |
| 2020-03-05 | 1.0.3 | 增加退款订单回调 |  |  |
| 2020-04-01 | 1.0.3 | 支付回调主币种字段增加注释 |  |  |
| 2021-11-11 | 1.0.4 | 修改服务端二次验证接口 |  |  |

**目录**

[1. 服务器端接口说明 3](#_Toc1833674080)

[1.1. 概要 3](#_Toc22444463)

[1.2. 协议说明 3](#_Toc1414451416)

[1.2.1 通信协议 3](#_Toc40976422)

[1.2.2 数据协议 3](#_Toc1495957514)

[1.3. 接口说明 3](#_Toc1966882369)

[1.3.1 用户会话验证接口（必接） 3](#_Toc1176197512)

[1.3.2 充值结果回调接口 5](#_Toc764613549)

[1.3.3 退款回调地址 8](#_Toc317774395)

[1.4. 签名生成 10](#_Toc42426676)

[1.4.1 签名生成方法说明 10](#_Toc100572728)

[1.4.2 签名校验工具 11](#_Toc256209307)

[1.4.3 常见错误 12](#_Toc405110514)

1.3.3 退款结果回调接口 12

[1.4. 签名生成 10](#_Toc1857994081)

[1.4.1 签名生成方法说明 10](#_Toc746808340)

[1.4.2 签名校验工具 11](#_Toc1713337312)

[1.4.3 常见错误 12](#_Toc451980161)

# 服务器端接口说明

## 概要

本部分主要提供“SDK服务器”和游戏合作商“游戏服务器”的交互的接口规范。

## 协议说明

### 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请求向“SDK服务器”发起接口请求。

### 数据协议

1. **数据格式**

该文档包括的所有接口，如无特殊说明，请求消息都是使用POST（application/x-www-form-urlencoded）格式提交数据，响应消息的内容为json格式数据，具体请参考下文的例子。

1. **字符编码**

请求与响应内容须采用utf-8字符编码。

## 接口说明

### 用户会话验证接口（必接）

1. **请求地址**

[https://krm.icebirdgame.com/user/](http://token.aiyinghun.com/user/)token/v2

1. **调用方式：**HTTP POST
2. **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相关数据

“游戏客户端”通过“SDK客户端”获取到sid，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，获取用户的账号信息。

1. **请求方：**游戏服务器
2. **响应方：**SDK服务器
3. **请求内容：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **字段说明** | **类型** | **必填** | **备注** |
| sid | Sid | string | Y | 登录后从渠道获取的sid |
| sign | 签名结果 | string | Y | 签名方法详见1.4 |

1. **返回内容（json格式）：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **字段说明** | **类型** | **必填** | **备注** |
| ret | 返回码 | Int | Y | 0标识成功，不为0标识失败 |
| msg | 返回信息 | String | Y | 成功或错误信息 |
| content | 响应数据 | json | Y | 包含data和cData两个json |
| **响应数据(对应content,采用json格式)包含的数据如下** | | | | |
| data | 返回数据 | json | Y | 内容见以下 |
| cData | 渠道返回数据 | json | Y |  |
| **响应数据(对应data,采用json格式)包含的数据如下** | | | | |
| gameId | 游戏ID | string | Y |  |
| channelId | 渠道ID | string | Y |  |
| appId | 游戏子包ID | string | Y |  |
| userId | 用户ID | string | Y | 验证后最终的uid |
| sdkData | sdk需要的数据 | json | Y |  |
| accessToken | 登录凭证 | string | Y |  |

**例子：**

{

"ret": 0,

"msg": "成功",

"content": {

"data": {

"gameId": "0001",

"channelId": "0001",

"appId": "0001",

"userId": "u0001",

"sdkData": {

"channelUid": "u0001"

},

"accessToken": "62d1e3be3d8a55cdabfc43786ebc325c"

},

"cData": []

}

}

### 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。游戏方下单时通过sdk将回调地址通过sdk规定的字段传递过来，不需要我方做任何回调地址的配置，详细参考sdk客户端文档。
2. **调用方式：**HTTP POST（application/x-www-form-urlencoded）
3. **接口描述：**

用户在游戏中提交充值请求后，sdk会到对应渠道去充值，在充值操作完成，接收到渠道回调的充值结果后，将充值结果再通知给“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。

1. **请求方：**SDK服务器
2. **响应方：**游戏服务器

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **字段说明** | **类型** | **必填** | **备注** |
| gameId | 游戏ID | string | Y |  |
| channelId | 渠道ID | string | Y |  |
| appId | 游戏包ID | string | Y |  |
| userId | 用户ID | string | Y |  |
| cpOrderId | 游戏方的订单ID | string | Y |  |
| bfOrderId | 订单ID | string | Y |  |
| channelOrderId | 渠道的订单ID | string | Y | 可能是空字符串 |
| money | 金额 | int | Y | 单位分 |
| callbackInfo |  | string | Y | 支付透参 |
| orderStatus | 订单状态 | int | Y | 0--支付失败  1—支付成功 |
| channelInfo | 渠道自定义信息 | string | Y | 目前不支持，固定为空字符串 |
| currency | 币种 | string | Y |  |
| product\_id | 商品id | string | Y |  |
| server\_id | 区服id | string | Y |  |
| game\_role\_id | 角色id | string | Y |  |
| time | 时间戳 | string | Y |  |
| sign | 签名 | string | Y | 签名算法见下文 |

6) 请求头携带参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **字段说明** | **类型** | **必填** | **备注** |
| X-BNPAY-SANDBOX | 沙箱状态 | string | Y | "1"表示沙箱；其他表示正式 |
| X-BNPAY-PAYTYPE | 支付方式 | string | Y | "mycard"表示mycard  "google"表示google-play支付  "mol"表示mol支付  "apple"表示苹果支付  “onestore”韩国onestore商店支付  “samsung”三星支付 |

**7) 返回内容（文本）：**

|  |  |
| --- | --- |
| **响应内容** | **响应内容描述** |
| {"ret": 0} | 成功，表示游戏服务器成功接收了该次充值结果通知  注意是0为成功 |
| {"ret": 1} | 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 |

**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，根据处理结果返回成功或失败。如果返回{"ret": 0}，则“SDK服务器”结束通知任务；如果返回

{"ret": 1}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。重复通知时间为1 3 7 10 20 30 40 50 60分钟这几个时间点，当一个小时候后还是不成功，则不再重试通知。

### 退款回调地址

**1）请求地址：**由游戏合作商提供。把地址发给我方运营，由我方运营配置到后台。

**2）调用方式：**HTTP POST（application/x-www-form-urlencoded）

**3）接口描述：**

用户退款成功后，将退款结果再通知给“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。

**4）请求方：**SDK服务器

**5）响应方：**游戏服务器

**字段说明：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **字段说明** | **类型** | **必填** | **备注** |
| bnOrderSn | 冰鸟订单号 | string | Y |  |
| cpOrderSn | 研发订单号 | string | Y |  |
| createTime | 创建时间 | string | Y |  |
| refundTime | 退款时间 | string | Y |  |
| sign | 签名 | string | Y |  |

**6) 返回内容（文本）：**

|  |  |
| --- | --- |
| **响应内容** | **响应内容描述** |
| {"ret": 0} | 成功，表示游戏服务器成功接收了该次退款结果通知  注意是0为成功 |
| {"ret": 1} | 失败，表示游戏服务器无法接收或识别该次退款结果通知，如：签名检验不正确、游戏服务器接收失败 |

**7) 通知机制：**

退款操作完成后， “SDK服务器”都会将退款结果通过“退款结果回调接口”发送到“游戏服务器”。“游戏服务器”收到“SDK服务器”的退款通知后，根据处理结果返回成功或失败。如果返回{"ret": 0}，则“SDK服务器”结束通知任务；如果返回

{"ret": 1}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。重复通知时间为1 3 7 10 20 30 40 50 60分钟这几个时间点，当一个小时候后还是不成功，则不再重试通知。

## 签名生成

### 签名生成方法说明

签名生成规则如下：

1. 将所有参数按键名进行字符串的字典序排序；
2. 将所有参数拼接成一个字符串，键和值之间使用“=”连接，键值对之间直接连接，不需要任何连接字符串。
3. 在所得字符串后拼接appSecret；
4. 对字符串进行MD5加密；

**例子（以下为演示值，具体参数值以游戏接入时的为准）：**

HTTP请求内容的值为：

gameId => 0001

channelId => 0001

appId => 0001

extra=> 0001

sid => 62d1e3be3d8a55cdabfc43786ebc325c

排序拼接后得出要签名的内容串为：

appId=0001channelId=0001extra=0001gameId=0001sid=62d1e3be3d8a55cdabfc43786ebc325c

假定app\_secret=202cb962234w4ers2aa，把app\_secret拼接到上述字符串后面，得到要进行MD5前的字符串：

appId=0001channelId=0001extra=0001gameId=0001sid=62d1e3be3d8a55cdabfc43786ebc325c**202cb962234w4ers2aa**

执行MD5得到签名结果为：

b13548a8bfcd30e1ce7252fc83432085

最后发送的请求参数的值为（使用POST的application/x-www-form-urlencoded格式发送）

gameId => 0001

channelId => 0001

appId => 0001

extra=> 0001

sid => 62d1e3be3d8a55cdabfc43786ebc325c

sign => b13548a8bfcd30e1ce7252fc83432085

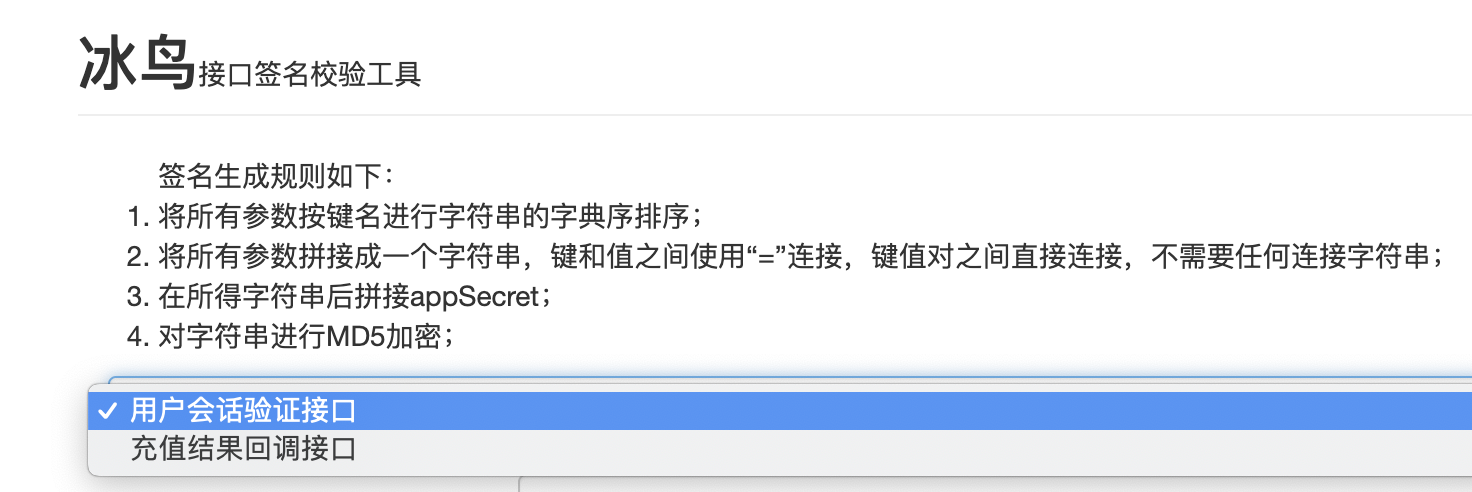
注意：

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)。

### 签名校验工具

工具地址： [https://krm.icebirdgame.com/fusion/verifySign](http://hwm.1aiyouxi.com/fusion/verifySign)

工具界面如下：



该工具主要是演示签名的生成过程，包括了服务端接口的两个主要接口：1.3.1和1.3.2，输入接口的参数和签名使用的密钥，即可生成正确的签名和显示签名生成的具体过程。

### 常见错误

检查md5前的字符串是否与演示工具中的一致， 如果不一致请检查是否字符串拼接的方式有误。如果一致请检查签名使用的密钥是否使用正确，如果对签名密钥有疑问，可以联系我方技术人员进行解答。