**谊游SDK服务端接入说明**

V 1.0.0

2015年8月11日

广州谊游网络科技有限公司

版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 说明 |
| V 1.0.0 | 2015-08-11 |  | 初稿 |
|  |  |  |  |
|  |  |  |  |

目录

[1. 流程 3](#_Toc432582545)

[1.1. SESSION校验流程 3](#_Toc432582546)

[1.2. 购买流程 4](#_Toc432582547)

[2. 接口通用说明 4](#_Toc432582548)

[2.1. 通信协议 4](#_Toc432582549)

[2.2. 数据格式 5](#_Toc432582550)

[2.3. 字符编码 5](#_Toc432582551)

[2.4. 签名 5](#_Toc432582552)

[2.5. 加密 7](#_Toc432582553)

[2.6. 会话 7](#_Toc432582554)

[2.7. ContextPath 7](#_Toc432582555)

[3. 用户类接口 7](#_Toc432582556)

[3.1. 登录状态查询 7](#_Toc432582557)

[4. 支付类接口 9](#_Toc432582558)

[4.1. 支付结果通知 9](#_Toc432582559)

本文档主要描述谊游游戏平台“SDK服务器”和游戏合作商“CP游戏服务器”的交互的接口规范。

## 流程

### SESSION校验流程



SDK登录成功后，CP 客户端即与CP服务器进行登录认证操作。在认证过程中，CP服务器需要通过SDK服务器校验客户端提交的session\_id是否有效。

### 购买流程



1. 用户发起购买请求后，首先CP客户端将订单信息提交到CP服务端创建订单，服务端将订单信息返回给客户端
2. 客户端调用SDK相关支付方法，SDK将引导用户完成支付
3. 支付成功后，SDK服务端会异步通知CP服务器进行发货
4. 本流程省略了SDK与支付中心的支付流程

## 接口通用说明

### 通信协议

SDK Server采用HTTP 协议作为通信协议，类REST风格，客户端通过构造HTTP请求，向服务器发起调用。

样例的请求如下：

|  |
| --- |
| POST /sdk\_callback/payback/181/1\_0\_0 HTTP/1.1  Host: www.xxx.cn  Content-Length: 223  sid：83504a24e0f77be482140e7c62b789d1  sign: 9W/nIm3WSvsmQzIQnpLppA==  {"cpOrderId":"2015153306030651","gameId":201508888,"payDate":"2015-11-25 18:21:34","payFee":"1.00","payResult":"1","sdkOrderId":"20151125182134550","uid":12345678} |

### 数据格式

请求消息和响应消息的内容都使用JSON格式。如：

|  |
| --- |
| {  "result":"0",  "message":"成功"  } |

传输时使用紧凑模式，即删除所有不必要的空白，以减少传输量。

### 字符编码

数据报文采用utf-8 字符编码。

### 签名

接口请求时，对请求报文（如：{"command":"add"}） + 密钥（如：1234567890）所得字串（如：{"command":"add"}1234567890）进行MD5摘要（一个byte数组），并对摘要作BASE64编码（如：4PIfnb1IQ+6gsbQoSKmD6w==），以编码作为请求签名，在请求时，以自定义的HTTP Header的方式一并发送，Header name为sign（如：sign：4PIfnb1IQ+6gsbQoSKmD6w== ）。

注意：

* 发送请求时，只传输报文，不传密钥；
* 密钥由线下方式提供；
* 响应不使用签名。

JAVA（JDK 7）样例代码：

|  |
| --- |
| **import** java.security.MessageDigest;  **import** sun.misc.BASE64Decoder;  **import** sun.misc.BASE64Encoder;  **public** **class** Main {  **public** **static** String encodeBASE64(**byte**[] key) **throws** Exception {  **return** (**new** BASE64Encoder()).encodeBuffer(key);  }  **public** **static** **byte**[] digestMD5(**byte**[] data) **throws** Exception {  MessageDigest md5 = MessageDigest.*getInstance*("MD5");  md5.update(data);  **return** md5.digest();  }  **public** **static** **void** main(String[] args) **throws** Exception {  String data = "{\"command\":\"add\"}";  String key = "1234567890";    String sign = *encodeBASE64*(*digestMD5*((data + key).getBytes("UTF-8"))).trim();  System.***out***.println(sign);  }  } |

设置自定义的HTTP Header：

|  |
| --- |
| String url = "https://selfsolve.apple.com/wcResults.do";  URL obj = **new** URL(url);  HttpsURLConnection con = (HttpsURLConnection) obj.openConnection();    con.setRequestMethod("POST");  //add reuqest header  con.setRequestProperty("User-Agent", "Mozilla/5.0");  con.setRequestProperty("Accept-Language", "en-US,en;q=0.5");  con.setRequestProperty("sign", "4hQWJd43m6k3rjyg21/aNQ=="); |

或者使用更加简单的Apache HttpClient

|  |
| --- |
| DefaultHttpClient httpclient = **new** DefaultHttpClient();  String url = "http://localhost";  HttpPost httpPost = **new** HttpPost(url);  httpPost.addHeader("sign", "4hQWJd43m6k3rjyg21/aNQ==");  HttpResponse response = httpclient.execute(httpPost); |

PHP可使用如下的方式：

|  |
| --- |
| base64\_encode(md5($str, true)) |

### 加密

暂不支持。

### 会话

用户登录后，服务器返回sid，以标识当次登录。

之后每请求时，在自定义的HTTP Header一并发送sid，Header name为sid（如：sid：83504a24e0f77be482140e7c62b789d1 ）。sid检验失败将返回用户未登录或会话已超时错误。

用户登出后，sid失效。

### ContextPath

线上测试环境：<http://120.132.68.148:18081/sdk.server/>

正式环境：http://sdk.52tt.com/sdk.server/

## 用户类接口

### 登录状态查询

#### 描述

查询用户是否已登录某游戏

注：登录时产生的会话ID，在HTTP Header中发送，系统将据SID来判断登录的状态，参考2.6节

#### 请求地址

[ContextPath]/rest/user/loginstatus.view

#### 请求方法

HTTP POST

#### 调用方向

SDK客户端 🡪 SDK Server；游戏服务端 🡪 SDK Server

#### 请求内容

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 父节点 | 限制 | 说明 |
| uid | long | - | 1 | 用户ID |
| gameId | Int | - | 1 | 游戏ID |

样例：

|  |
| --- |
| {  "uid":234721,  "gameId":225  } |

#### 响应内容

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 父节点 | 限制 | 说明 |
| head | Object | - | 1 |  |
| result | String | head | 1 | 返回代码  0为用户处于登录状态，其它为失败 |
| message | String | head | 1 | 说明 |

样例：

|  |
| --- |
| {  "head":{  "result":"0",  "message":"用户处于登录状态"  }  } |

## 支付类接口

### 支付结果通知

#### 描述

用户完成支付后，通知CP方发货

如果第一次调用失败，会重复多次调用该接口。系统可能会重复通知，商户必须保证对于多次重复调用与一次调用的结果是一致的。游戏服务端需要作参数签名验证，否则存在安全风险。

#### 请求地址

[由CP提供]

#### 请求方法

HTTP POST

#### 调用方向

SDK Server 🡪 CP Server

#### 请求内容

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 父节点 | 限制 | 说明 |
| uid | long |  | 0..1 | 用户ID |
| gameId | Int |  | 0..1 | 游戏ID |
| sdkOrderId | String | - | 0..1 | SDK订单号 |
| cpOrderId | String | - | 0..1 | CP订单号 |
| payFee | String | - | 0..1 | 实际支付金额，单位：元 |
| payResult | String | - | 0..1 | 支付结果 1成功，其它失败 |
| payDate | String | - | 0..1 | 支付时间，格式'yyyy-mm-dd hh:mm:ss' |
| exInfo | String |  | 0..1 | CP扩展信息 |

样例：

|  |
| --- |
| {  "uid":20150515213406829,  "gameId":20150515213406829,  "sdkOrderId":"20150515213406829",  "cpOrderId":"20150515213406829",  "payFee":"16.00",  "payResult":"1",  "payDate":"2015-05-15 15:27:20",  "exInfo":"abc"  } |

#### 响应内容

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 父节点 | 限制 | 说明 |
| head | Object | - | 1 |  |
| result | String | head | 1 | 返回代码  0为成功，其它为失败 |
| message | String | head | 1 | 说明 |

样例：

|  |
| --- |
| {  "head":{  "result":"0",  "message":"成功"  }  } |