Nubia应用中心支付SDK接入指南

**服务端**

# 版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 日期 | 作者 | 修订要点 |
| V1.0.0 | 2015-12-5 | 顾辉 | 添加 |
| V1.0.1 | 2015-12-11 | 罗川 | 3.1添加接口地址，添加响应中的字段retCode=3表示未登录  3.2的响应中添加字段orderStatus  增加签名说明 |
| V1.0.2 | 2015-12-29 | 顾辉 | 添加正式域名 |
| V1.0.3 | 2016-2-25 | 顾辉 | 3.2的响应中添加字段gameId、propId、propUnit、propPrice |
| V1.0.4 | 2016-4-7 | 顾辉 | 添加签名规则2.4  3.1 的请求字段中添加data\_timestamp，修改签名规则。  3.2 将“验证订单信息”改为“CP服务端订单生成接口说明”  3.3 请求字段中添加data\_timestamp，修改响应字段和签名规则。 |
|  | 2016-4-10 | 罗川 | 添加订单签名的示例代码3.4 |
| V2.0.0 | 2016-8-8 | 罗川 | 变更联运的域名，并且采用https |
|  |  |  | game\_id不参与签名，但是必填 |

**目录**

[Nubia应用中心支付SDK接入指南 1](#_Toc448251814)

[版本记录 2](#_Toc448251815)

[一、接入概述 4](#_Toc448251816)

[1.1 申请相关参数 4](#_Toc448251817)

[1.2 SESSION校验流程 4](#_Toc448251818)

[1.3 支付流程说明 4](#_Toc448251819)

[二、协议说明 6](#_Toc448251820)

[2.1 域名 6](#_Toc448251821)

[2.2 通信协议 6](#_Toc448251822)

[2.3 数据协议 6](#_Toc448251823)

[2.4 签名规则 7](#_Toc448251824)

[三、接口说明 8](#_Toc448251825)

[3.1 SESSION校验 8](#_Toc448251826)

[3.2 CP服务端订单生成接口说明 9](#_Toc448251827)

[3.3 通知游戏服务器支付情况 10](#_Toc448251828)

[3.4 签名算法示例 11](#_Toc448251829)

# 一、接入概述

## 1.1 申请相关参数

用户需要向我司开发者平台申请AppID，AppKey，AppSecret

AppID：App的唯一ID

AppKey：App的唯一标识

AppSecret：App的秘钥，用于服务器之间的签名

否配对。相关接口说明见3.1。

## 1.2 通信协议

支付SDK服务器采用HTTP协议作为通信协议。可以通过构造HTTP请求（POST方式）向支付SDK 服务器发起接口请求。而返回结果统一以JSON数据格式返回。

**格式统一如下：**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 必选 | 字段类型 | 描述 |
| code | Y | int | 状态（0：成功；其余为失败） |
| data | Y | Json数组或Json对象 | 数据或列表 |
| message | N | String | 消息（状态的中文描述），失败必须填写原因 |

**返回示例：**

|  |
| --- |
| {  "code":0, //必选，返回码 （0为成功，其余为失败）  "data":{}, //必选，返回的数据  "message":"" //可选，返回消息，出现错误时使用此消息展示给用户  } |

## 1.3 数据协议

1) 数据格式

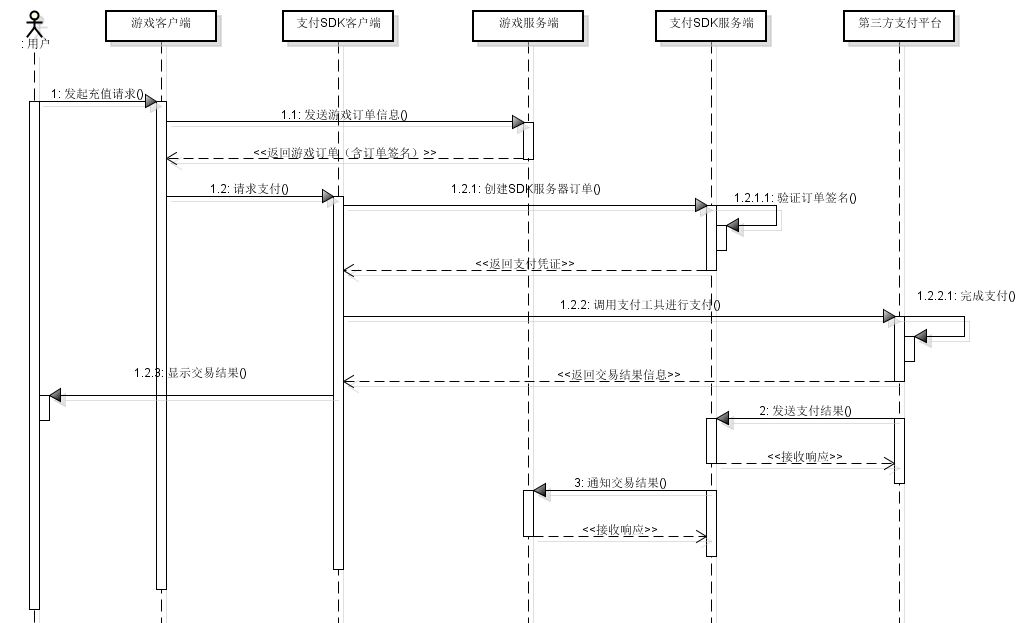
请求信息采用HTTP 协议Key-Value 方式提交，响应消息的内容使用JSON格式表示数据，具体请参考下文的示例。

2) 字符编码

请求与响应内容须采用UTF-8字符编码。

# 二、支付服务端对接

## 2.1 支付流程说明

****

1、用户发起充值请求后，首先游戏客户端向游戏服务器端发送游戏订单的相关信息，游戏服务端会返回一个游戏订单信息（包含订单签名）

2、游戏客户端收到游戏订单信息后，向支付SDK客户端发送请求支付的信息，其中包含了游戏订单签名

3、支付SDK客户端收到游戏客户端发送的支付请求后，向支付SDK服务端请求创建支付订单。而支付SDK服务端收到请求后会首先验证游戏订单签名是否正确，如果正确会返回一个支付凭证，用于支付SDK客户端向第三方支付平台（如：支付宝服务器）发送支付请求

4、支付SDK客户端收到支付凭证后，调用第三方支付工具进行支付。第三方支付平台（如：支付宝服务器）完成相应的支付操作后向支付SDK客户端返回交易结果信息，同时，异步向支付SDK服务端通知支付结果

5、支付SDK服务端收到交易结果后，通知游戏服务端交易结果。而游戏服务端收到交易结果后需要告知支付SDK服务端已收到

## 2.2 cp服务器订单产生需要的参数和签名

**cp服务器订单需要产生下面的参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **字段类型** | **参与签名** | **必填** | **字段说明** |
| app\_id | String | Y | Y | 应用id |
| uid | String | Y | Y | 登录时获取的uid，该uid必须和nubia账号相关联 |
| cp\_order\_id | String | Y | Y | 订单号(长度限制为50个字符,由数字、英文字母、”\_”和”-”组成) |
| amount | BigDecimal | Y | Y | 支付金额（精确到分,单位为元） |
| product\_name | String | Y | Y | 商品名称 |
| product\_des | String | Y | Y | 商品描述 |
| number | int | Y | Y | 商品数量 |
| data\_timestamp | String | Y | Y | 订单产生的时间戳 |
| game\_id | String | N | Y | 玩家小号（不参与签名，但是必填） |
| product\_id | String | N | N | 商品ID（不参与签名） |
| product\_unit | String | N | N | 商品单位（不参与签名） |
| product\_price | String | N | N | 商品单价（不参与签名） |
| cp\_order\_sign | String | N | Y | cp服务器产生的订单签名 |

**上述标红的字段为参与签名的参数，参与签名的参数不可为空，签名规则见2.4。**

**如果商品名称和商品描述需要特殊字符，只允许传 ! @ # $ ^ & \* ( ) \_ ; ' : < > , . ? / | 特殊字符**

## 签名规则

1. 参数排序：将所有签名参数按字母顺序进行排序, 若遇到相同首字母，则看第二个字母。

2. 参与签名的数据不要做URL Encoding，一律以UTF-8编码参与签名。

3. 参与签名的数据中，必须有字段data\_timestamp，为接口发起的unix时间戳，如果传入的时间与服务器时间相差5分钟以上，服务器可能会拒绝请求。data\_timestamp在linux服务器上是命令date +%s获得的时间，也就是从1970-01-01 00:00:00 UTC起，到当前时间的秒数。

4. 参数拼接：将所有参数按k1=v1&k2=v2&k3=v3…格式进行拼接，不包含sign。

5. 签名：将第4步所得字符串+”:”+AppID+”:”+ AppSecret用md5计算摘要，得签名字符串，产生的**签名字符串是小写格式**。所有的签名示例可以见附件的代码。

签名字符串示例：

|  |
| --- |
| amount=1.00&app\_id=15015&cp\_order\_id=nb1451635941847&data\_timestamp=1458824465&number=1&product\_des=牛币可以交易&product\_name=牛币&uid=150:appId:secret\_key |

产生的签名值给cp\_order\_sign

## 2.3 通知cp订单支付情况

支付组件服务器接收到第三方支付服务端（如：支付宝服务端）返回的支付通知后，会向各业务服务器通知订单的支付情况。因而，各业务必须提供支付结果通知地址notifyUrl（在开发者平台上注册申请appId和appKey时提供）。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 通知支付情况 | | |  | | 调用方式 | | HTTP Post | |
|  | | 请求方 | | 支付组件服务端 | |
|  | | 响应方 | | 业务服务端 | |
|  | | 同步or异步 | | 同步 | |
|  | 请求内容 | | | | | | | |
| 字段名称 | | 字段类型 | | 参与签名 | | 必填 | | 字段说明 |
| order\_no | | string | | Y | | Y | | 游戏订单号 |
| data\_timestamp | | string | | Y | | Y | | 支付通知时间 |
| pay\_success | | int | | Y | | Y | | 支付成功与否(成功为1) |
| sign | | string | | N | | Y | | 支付通知签名（包含参数order\_no、data\_timestamp，具体规则见2.4，该签名为旧版使用，新接入的应用统一使用order\_sign来校验） |
| app\_id | | String | | Y | | Y | | 应用id |
| uid | | String | | Y | | Y | | 登录时获取的uid，该uid必须和nubia账号相关联 |
| amount | | BigDecimal | | Y | | Y | | 支付金额（精确到分） |
| product\_name | | String | | Y | | Y | | 商品名称 |
| product\_des | | String | | Y | | Y | | 商品描述 |
| number | | int | | Y | | Y | | 商品数量 |
| order\_serial | | Int | | N | | N | | Nubia支付的订单号 |
| order\_sign | | string | | N | | Y | | 支付通知签名（包含参数order\_no、data\_timestamp、pay\_success、app\_id、uid、amount、product\_name、product\_des、number，具体规则见2.4，新接入的应用统一使用order\_sign来校验） |

各业务服务端必须对支付通知签名进行验证，支付通知签名包含order\_no、data\_timestamp、pay\_success、app\_id、uid、amount、product\_name、product\_des、number ，规则同2.3。验证成功后，进行支付完毕后的相关处理（例如：发道具等），并进行如下响应，以示收到支付通知。

注意上面的sign和order\_sign参数，sign是支付组件老版本的签名，order\_sign是新版本的签名，二者必填一个就可以了，这两个签名所需的签名参数也不一样。建议使用order\_sign。

如果商品名称和商品描述需要特殊字符，只允许传 ! @ # $ ^ & \* ( ) \_ ; ' : < > , . ? / | 特殊字符

**响应：**

|  |  |  |  |
| --- | --- | --- | --- |
| 响应内容： | | | |
| 协议：http+json | | | |
| 字段名称 | 字段类型 | 必填 | 字段说明 |
| code | int | Y | 0：成功；  10000：发货失败；  90000：其他异常 |
| data | Json | Y | 暂可为空 |
| message | string | N | 失败时必填，请填写失败的原因；  成功可以不用填写 |

**响应示例：**

|  |
| --- |
| {  "code": 0,  "data": {},  "message": "成功"  } |

|  |
| --- |
| {  "code": 10000,  "data": {},  "message ": "失败的原因"  } |

响应部分极为重要，nubia服务端会监听此响应信息并有对应预警邮件发送至渠道运营，监控cp掉单情况

# 三、用户登录信息验证接口

## 3.1 SESSION校验流程



SDK登录成功后，游戏客户端即与游戏服务器进行登录认证操作。在认证过程中，游戏服务器需要通过SDK服务器校验客户端提交的uid与sessionId是各业务服务器需要根据各自业务的需要进行用户的登录认证，若应用允许不登陆也可支付请忽略本步骤。

校验接口地址：

<https://niugamecenter.nubia.com/VerifyAccount/CheckLogined>

联运sdk登录验证参数如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 通知支付情况 | | 调用方式 | | | HTTP Post | |
| 请求方 | | | 业务服务端 | |
| 响应方 | | | 支付组件服务端 | |
| 同步or异步 | | | 同步 | |
| 请求内容 | | | | | | |
| 字段名称 | 字段类型 | | 参与签名 | 必填 | | 字段说明 |
| uid | string | | Y | Y | | 用户uid，不用于映射游戏内账号 |
| data\_timestamp | string | | Y | Y | | 接口请求时间 |
| game\_id | string | | Y | Y | | 用户的游戏id，用于标识用户唯一身份，映射游戏账号 |
| session\_id | string | | Y | Y | | sessionId |
| sign | string | | N | Y | | 签名（具体规则同2.2） |

**响应：**

|  |  |  |  |
| --- | --- | --- | --- |
| 响应内容： | | | |
| 协议：http+json | | | |
| 字段名称 | 字段类型 | 必填 | 字段说明 |
| code | int | Y | 0：成功；  3: 未登录  7:参数有为空  10: 校验失败,签名错误  14: 校验失败，uid不匹配  15: 校验失败，game\_id不匹配 |
| data | Json | Y | 暂可为空 |
| message | string | N | 失败时必填，请填写失败的原因；  成功可以不用填写 |

**响应示例：**

|  |
| --- |
| {  "code": 0,  "data": {},  "message": "成功"  } |

|  |
| --- |
| {  "code": 3,  "data": {},  "message ": "session过期或者没有登录"  } |

## 3.2 签名算法示例

请看附件sign 工程，其中有对签名的示例