

# SDK-Server接入指南

## 一、概述

本文档用于CP快速接入服务器端SDK-Server

### 接口介绍

接口名	请求URI	功能描述	结果说明
token验证接口	正式服: http://rhs.palmfungames.com/integration/api/verifyToken 测试服: http://120.92.173.43:8080/integration/api/verifyToken	玩家登录获取token到SDK服务器验证token是否有效。	详细见：Token验证接口
支付回调接口	( CP提供)支付回调地址	该接口用户支付后通知商户服务器	后台在notify_url后添加参数返回，要求后台notify_url收到通知后进行响应。

## 二、接口接入详情

### 1. token校验接口

- 请求URL地址:见上表
- 请求方式:post/get
- 请求参数

参数名称	类型	事例值	是否必须	描述
appKey	String		是	我方提供
token	String		是	用户验证的token
sign	String		是	MD5(token+appKey+appSecret)

- 返回参数以json的格式返回

参数名称	类型	事例值	描述
code	int	200	返回200代表成功
error_msg	String		错误的信息
data.userId	String		用户ID(平台用户唯一id)
data.thirdId	String		第三方用户Id (渠道用户Id)

请求例子:

```
{“sign”:“bf9d4d8bee4b5163be52e159b865d731”,“token”:“3DE6B950047D4244A9021F0200E1EB29”,“appKey”:“F50960947EBAC497DA14172A44C899D0”}
```

返回结果:

```
result:{“code”:“200”,“error_msg”:“成功”,“data”:{“userId”:“1000819587”,“thirdId”:“B7E8F4223732778BCA3CEFF1E6E3CD12”}}
```

## 2.充值回调结果接口

- 请求URL地址:
- 请求方式:post/get
- 请求参数

参数名称	类型	事例值	是否必须	描述
transactionId	String		是	交易流水号（平台唯一）
appId	int	10003	是	由平台提供，应用ID
roleId	String		是	游戏角色ID
gameOrderId	String		是	游戏订单ID
gameZoneId	String		是	游戏区服ID
productId	int		是	平台提供的商品ID
productName	String		是	游戏商品名称
money	String		是	商品价格(单位：元)
status	int		是	订单状态:0-订单初始化,1-成功,4-失败
cpPrivateInfo	String		否	游戏下订单时传给SDK的rExInfo
channel	int		否	第三方渠道号
model	String		否	如果是sandbox表示为测试的沙盒订单，不用作结算；正常订单没有此字段
sign	String		是	md5(gameOrderId+productId+transactionId+appKey+appSecret)

- 返回参数以json的格式返回

参数名称	类型	事例值	描述
------	----	-----	----

code	int	200	返回200代表成功，其他表示失败！
error_msg	String		错误的信息

返回例子结果：

result:{"code": "200", "error\_msg": "成功"}

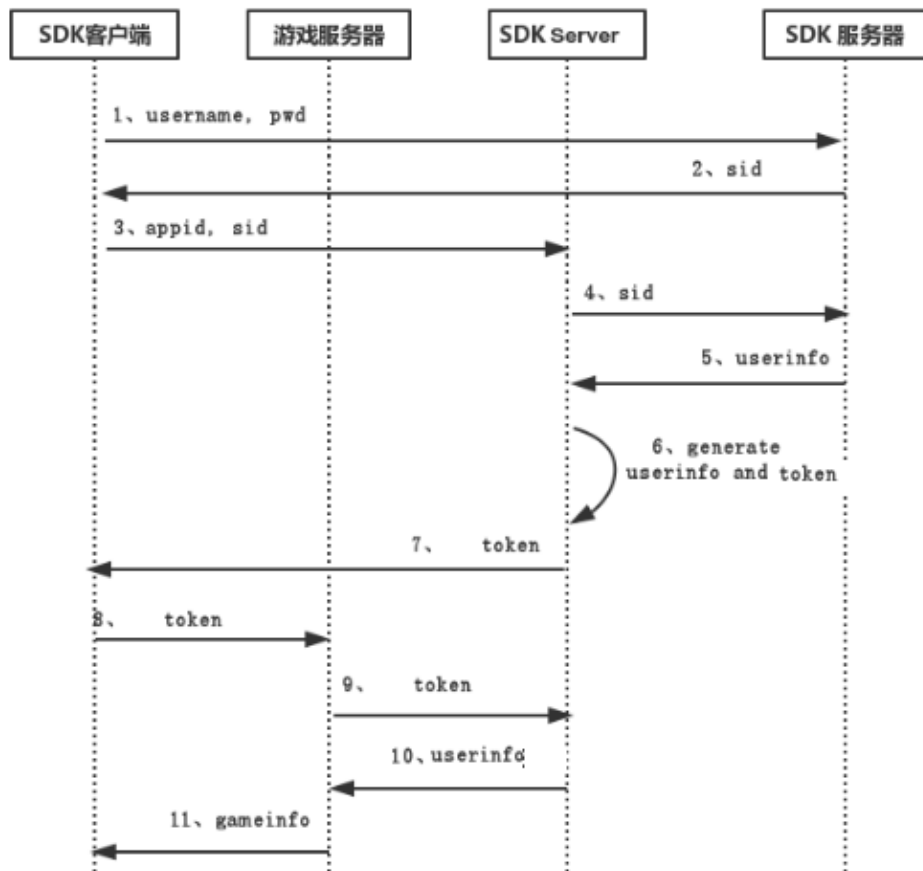
注：失败的订单会间隔5分钟在次通知，如果一直不成功会累计通知5次，5次以上将不在通知

---

## 三、业务流程

### 1、登录注册流程

- 登陆认证的时序图:

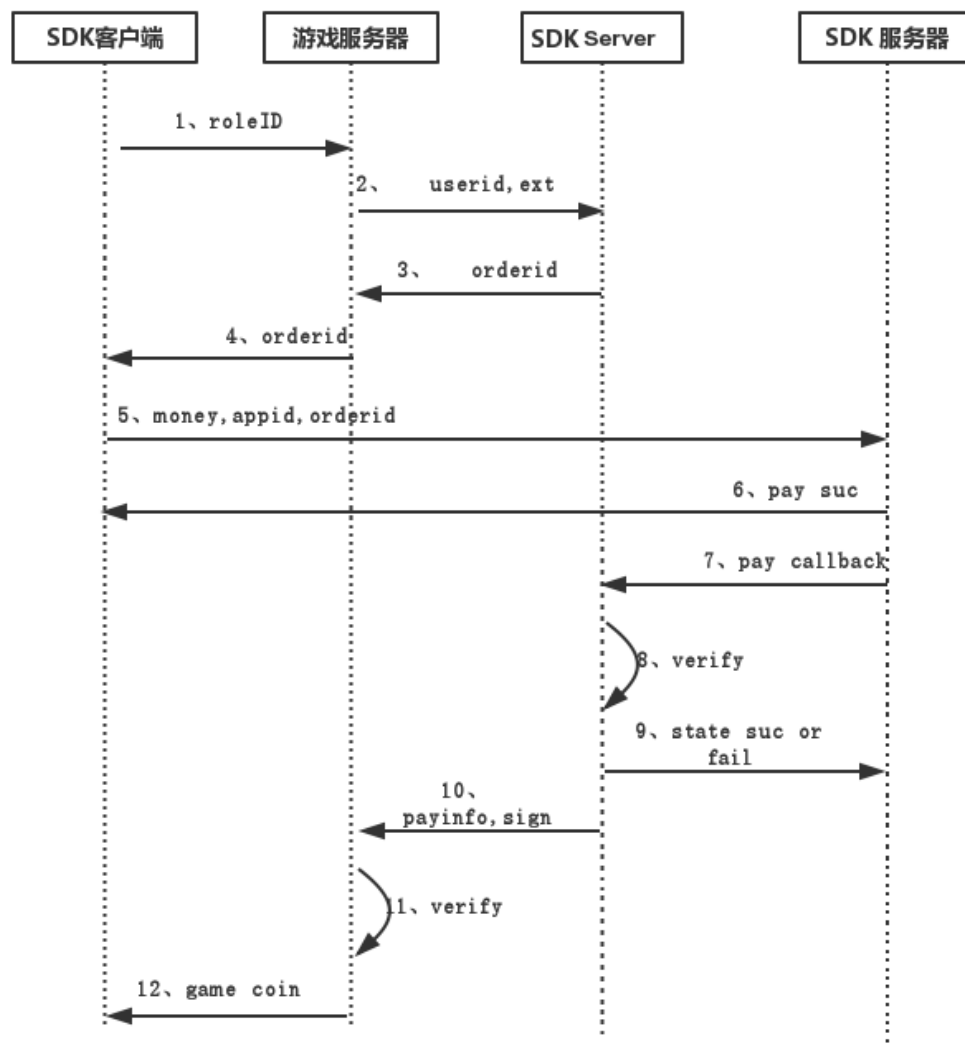


- 1、客户端接入抽象SDK框架，根据当前具体是哪个SDK渠道，调用登陆界面，然后传入用户名和密码，进行SDK登陆操作
- 2、SDK登陆成功，会返回sid等信息
- 3、游戏客户端可以通过SDK抽象层的接口，获取到这个sid
- 4、游戏客户端拿着这个sid以及接入之前向 server申请的appid,渠道号等信息，Http访问server(游戏验证server)进行登陆认证。
- 5、游戏验证server 根据当前传递的appid, 渠道号，去对应的SDK服务器进行认证
- 6、SDK服务器认证成功，会返回SDK服务器那边的用户信息
- 7、游戏验证 Server拿到用户信息，生成一个游戏验证server统一的用户信息并存储。然后，紧接着返回给客户端一个有效的token。
- 8、客户端拿着这个token，去访问游戏服务器(多数是游戏登陆服务器)
- 9、游戏服务器，拿着这个token去游戏验证 server 进行登陆认证。

10、游戏验证 server 判定token有效，则返回给游戏服务器当前用户的用户信息

11、游戏服务器拿到用户信息，证明当前登陆成功，返回给客户端服务器列表等数据，登陆成功。

## 2、支付流程



- 1、游戏客户端，首先请求游戏服务器要充值
- 2、游戏服务器拿着该用户的id和一些支付成功之后需要原样返回的数据，去访问游戏验证 Server申请订单号
- 3、游戏验证 Server生成一个唯一的订单号，同时数据库中生成一条订单记录，状态是正在支付状态
- 4、游戏服务器将订单号返回给客户端
- 5、游戏客户端，拿到订单号之后，带着订单号以及游戏里充值相关的数据，调用SDK抽象接口的支付接口，调用对应的SDK支付界面，进行充值操作。
- 6、当前SDK的渠道实现在调用SDK支付界面之前，需要把刚刚的订单号，放到渠道SDK支付参数的自定义参数中。这个每个渠道都是一样的。
- 7、渠道SDK支付成功，立马返回一个状态
- 8、同时，渠道SDK服务器会异步通知游戏开发商设置的支付回调地址。注意，游戏接入的时候，这个回调地址要设置到旋游戏验证 server提供的一个地址。
- 9、游戏验证server收到充值回调，根据验证结果等判定，立马给渠道SDK服务器返回一个成功或者失败的状态。
- 10、然后游戏验证 server根据自定义参数中的orderId，查询到对应的订单信息，再根据订单信息，获取到当前用户信息和对应的游戏信息，然后调用接入  
游戏之前，游戏服务器提供给旋游戏验证server的支付回调地址。这个回调地址，游戏服务器只需要提供一个给 游戏验证server就可以了。因为游戏服务器只和 游戏验证 server交互。
- 11、游戏服务器收到回调，验证成功与否，里面返回给游戏验证server一个成功或者失败的信息。同时，给对应的玩家加游戏币。