

Global SDK 服务器端接入文档

SDK 服务器域名

全球地区: <https://mapi.perfectworldgames.com>

通用规则

请求数据格式

Content-Type 如下

```
1 Content-Type: application/x-www-form-urlencoded
```

注意事项

1. http 中的数据 (url 和 body 中的参数) 不保证顺序, SDK 服务器和游戏服务器, 在接收到数据后, 需要自行排序后再使用签名规则

签名规则

1. 签名算法使用 RSA 算法
2. 游戏服务器调用 SDK 服务器使用 一套 RSA 密钥, 由游戏服务器生成并将公钥字符串提供给 SDK 服务器, 密钥编码为 base64
3. SDK 服务器通知游戏服务器使用另一套 RSA 密钥, 由 SDK 服务器生成并将公钥字符串提供给游戏服务器, 密钥编码为 base64
4. 待签名字符串的生成:
 - 4.1. sign 参数不需要参与签名
 - 4.2. 非必选参数如果出现在 http 请求中, 则必须参与签名
 - 4.3. 将所有参与签名的参数按照参数名的字母顺序升序排列, 如 appId: 1, productName: 商品名, uid: 2
 - 4.4. 将排序后的参数拼接成 key1=value1&key2=value2&key3=value3&..... 的字符串 srcStr, 如 appId=1&productName=商品名&uid=2

5. 发送接口生成签名:(游戏服务器提供 RSA 公钥给 SDK 服务器)
- 5.1. 将 srcStr 和游戏服务器私钥 privateKey 进行 SHA1WithRSA 签名 signBytes
 - 5.2. 将 signBytes 进行 base64 编码得到最终的签名字符串
6. 接收接口验证签名:(SDK 服务器提供 RSA 公钥给游戏服务器)
- 6.1. 将接收到的签名 sign 进行 base64 解码成 byte 数组 signBytes
 - 6.2. 将 srcStr、signBytes 和 SDK 服务器公钥 publicKey 进行 SHA1WithRSA 签名验证
7. 接口文档中的字段可能会增加，生成签名数据时，请勿只使用当前文档中的字段，而应该使用接口中实际传递的所有字段

Global SDK 服务端接口

检查 token 是否有效【必接】

提供方

SDK 服务器提供该接口

请求地址

/s/api/game/user/token/check

请求方式

POST

请求参数

字段名	类型	必选	说明	取值范围	示例	备注
appld	long	y	应用在 SDK 平台的唯一标识			
t	long	y	当前时间戳（毫秒）			
token	string	y				
uid	long	y	用户在 SDK 服务器的唯一标识，客户端获取			

sign	string	y				
serverId	string	n				
osType	long	n		<ul style="list-style-type: none">● 2: ANDROID● 3: IOS		
version	Integer	n	当此字段值传 2 时，返回数据中会增加 whiteUser 字段,用来区分是否为白名单用户			

http res body 结构

字段名	类型	必选	说明	取值范围	示例	备注
code	long					
result	JSON	n				

result 结构

字段名	类型	必选	说明	取值范围	实例	备注
whiteUser	Integer	n		<ul style="list-style-type: none">● 0: 非白名单用户● 1: 白名单用户		

code 说明

code 取值	说明
0	操作成功

返回示例

```
1 {
2   "code":0
3 }
```

储值成功回调游戏服务器【必接】

提供方

游戏服务器提供该接口

请求地址

- 1. 可以通过游戏客户端下单传递划拨地址
- 2. 在管理后台配置划拨地址

以上两个方式都可以，游戏任意选择

请求方式

POST

请求参数

字段名	类型	必选	说明	取值范围
uid	string	y	用户在 SDK 服务器的唯一标识	
appld	long	y	应用在 SDK 平台的唯一标识	
sdkOrderId	string	y	<ul style="list-style-type: none">SDK 平台的订单号订阅型商品自动续订成功时，sdkOrderId 会新创建，与首次订阅的 sdkOrderId 不同订阅型商品取消续订时，sdkOrderId 同首次订阅一样	
subscribeSdkOrderId	string	n	<ul style="list-style-type: none">如果该值存在，说明这个是根据之前的一个订阅订单，自动生成的订单取值为之前订阅的一个 sdkOrderId	
t	long	y	当前时间戳（毫秒）	

appOrderId	string	n	应用在游戏服务器的订单号，对应客户端文档的 gameId	
moneyAmount	long	y	<ul style="list-style-type: none"> ● 实际支付金额 ● 从 SDK 客户端获取 ● 具体值未知，可能随着用户的支付账号、支付地区、支付方式变化 	
moneyCurrency	string	y	<ul style="list-style-type: none"> ● 实际支付币种 ● 从 SDK 客户端获取 ● 具体值未知，可能随着用户的支付账号、支付地区、支付方式变化 	
orderAmount	long	y	<ul style="list-style-type: none"> ● SDK 服务器后台配置的金额 ● 配合 payType 和 productId 可用于游戏服务器检验订单金额 	
orderCurrency	string	y	<ul style="list-style-type: none"> ● SDK 服务器后台配置的币种 ● 配合 payType 和 productId 可用于游戏服务器检验订单金额 	
serverId	string	y	游戏服务器 ID	
roleId	string	y	游戏角色 ID	
payType	long	y		<ul style="list-style-type: none"> ● 1: GooglePay ● 2: AppStore ● 3: TStore ● 4: KaKaoGameShop ● 5: WeixinPay ● 6: AliPay ● 7: Paypal ● 11: OneStore ● 12: SheenPay

				<ul style="list-style-type: none"> ● 13: MyCardPay ● 14: HuaWeiPay ● 15: DirectnessPay ● 16: PagsmilePay ● 17: RazerPay ● 18: CodaShopPay ● 19: Flexion ● 20: Razershop ● 21: HonorPay ● 22: XsollaPay ● 23: Payletter ● 24: Samsung ● 26: Steam ● 27: PlayStation ● 28: RuStore ● 29: 空中云汇 Airwallex
productId	string	y	商品 ID	
productName	string	n	商品名	
channelName	string	n	苹果、google 等渠道名的缩写	
channelId	string	n	苹果、google 等渠道 ID	
channelOrderId	string	y	苹果、google 等渠道产生的订单号	
sandbox	boolean	y		<ul style="list-style-type: none"> ● true: 沙箱订单 ● false: 正式订单
subscribe	boolean	y	该商品是否为订阅商品。如果是订阅商品，不论 unsubscribe 取何值，	<ul style="list-style-type: none"> ● true: 订阅订单 ● false: 非订阅订单

			subscribe 始终为 true	
unsubscribe	boolean	n	当 subscribe 为 true 时， unsubscribe 的值才需要考虑	● true: 取消订阅
platformId	long	y		● 2: ANDROID ● 3: IOS ● 4: windows ● 6: web 端储值
sign	string	y		
appExtraInfo	string	n	● 游戏透传字段 ● 格式为 json ● sdkServerLog 为保留的 key 前缀	
sdkExtraInfo	string	n	格式为 json	

sdkExtraInfo 结构

字段名	类型	必选	说明	取值范围	实例	备注
sdkChannelId	String	n		产品提供		
sdkMediaId	String	n		产品提供		

http res body 结构

字段名	类型	必选	说明	取值范围	示例	备注
code	long					

code 说明

code 取值	说明
---------	----

0	<p>操作成功</p> <ul style="list-style-type: none"> ● 该接口可能由于网络等问题，重复调用 ● 相同订单 sdkOrderId 重复通知时，需要返回 code 0 ● 除了返回 code 0，其他所有情况都算失败，包括网络失败，SDK 服务器会重新划拨 ● 游戏服务器需做防重处理
---	---

返回示例

```

1 {
2   "code":0
3 }
```

说明：

1. 客户端充值成功之后，所有元宝的划拨要以服务器端接口为准，不能以客户端为准
2. 服务器端的订单号，需要根据 sdk 的订单号 sdkOrderId 排重，即相同的 sdkOrderId 只能发一次元宝，sdk 可能由于网络原因发起多次划拨
3. 储值成功，SDK 这边会按照 2 的指数进行多次划拨，直到游戏服务器返回{code:0},由于网络原因返回失败，则会再次划拨。
4. 划拨多次，如果游戏 sdkOrderId 已经划拨成功，则不给游戏元宝，但是需要返回成功{code:0}
5. 对于限购商品，例如月卡的逻辑，由于玩家有限购次数，但是所有的储值又是异步，有可能造成玩家多次支付，对于这种问题，SDK 给了如下方案，第一，sdk 后台可以按照 productid 配置购买频次，即在玩家 sdk 支付成功到游戏划拨成功之前，sdk 不允许玩家二次购买，这样可以减少多次购买的问题。第二，如果玩家真的购买成功了，游戏需要给玩家第二次购买的转换成元宝
6. 游戏的订单号，目前只能作为一个附带功能，sdk 不做任何处理，只转发，由于 web 端充值，这个游戏订单号可能为空，不建议游戏在使用这个检验
7. moneyAmount 和 moneyCurrency 对应玩家的真实币种充值，游戏服务器可以只记录，不做真实校验
8. orderAmount 这个是 sdk 后台和游戏后台配置表中对应的每个档位的真实币种，需要游戏校验
9. 游戏服务器端需要校验的内容，uid、角色 id、productid、orderAmount 储值金额、sdkOrderId，必须相符在给划拨
10. payType 对应的支付类型，游戏的 addcash 打点可以使用

Global SDK server 出口 ip

如果游戏 server 有防火墙，需要开通以下 ip 的防火墙规则

全球地区：

```
1 47.254.41.89
2 47.254.41.182
3 47.89.225.76
4 47.254.78.63
5 47.254.35.48
6 47.254.64.58
7 47.254.90.133
8 47.88.5.229
9 47.88.58.9
```

常见错误处理

- code 10002: 缺少参数
- code 10011: 参数类型不对
- code 10004: 参数 t 的范围不对，应当使用当前的标准时间，预防重放攻击
- 调用接口 `/s/api/game/echo`，通过返回值，查看提交给 SDK 服务器的参数
- code 10003: 签名字段 sign 错误
 - ¢ 5.1. 有些框架会对提交的数据，自动 urlencode，导致数据二次 urlencode。
调用接口 `/s/api/game/echo`，通过返回值，查看提交给 SDK 服务器的参数
 - ¢ 5.2. 如果参数没有二次 urlencode 的问题，换个私钥（长的是私钥，短的是公钥）试下
 - ¢ 5.3. 检查游戏服务器签名前的字符串
 - 5.3.1. 是否把 sign 字段也签名了
 - 5.3.2. 是否缺参数
 - 5.3.3. 字段顺序是否正确
 - ¢ 5.4. 提供信息给 SDK 服务器开发人员

- 5.4.1. SDK 服务器返回的 message
- 5.4.2. 游戏服务器签名前的字符串（注意是一个字符串，最终签名的那个字符串，SDK 服务器用于线下比对签名结果）
- 5.4.3. 游戏服务器签名的结果