# 支付中心白皮书

V2.0

编制历史

|  |  |  |
| --- | --- | --- |
| 日期 | 操作人员 | 说明 |
| 2021-3-28 | Rossen | 重写支付中心白书皮 |
| 2021-4-28 | Rossen | 扩充2.1协议，使支持捕鱼产品 |
| 2021-7-1 | Rossen | 充值请求、支付返回游服等数据调整 |
|  |  |  |
|  |  |  |

目录

[支付中心白皮书 1](#_Toc15550)

[一、 概述 4](#_Toc2109)

[1.1、专用名词 4](#_Toc27581)

[1.2、签名规则 4](#_Toc5686)

[1.2.1 支付中心签名规则 4](#_Toc25359)

[1.2.2 游服签名规则 4](#_Toc30245)

[1.3、 5](#_Toc29872)

[二、 支付API 5](#_Toc9116)

[2.1、查询可用的支付渠道 5](#_Toc24928)

[2.2、发起支付 6](#_Toc10727)

[2.3、支付回调 7](#_Toc26249)

[2.3.1 google支付回调 7](#_Toc23279)

[2.3.2 apple支付回调 7](#_Toc30728)

[2.3.3 其他支付平台回调 8](#_Toc1615)

[2.3.4 通知游服支付结果 8](#_Toc28022)

[2.4、发起提现 9](#_Toc31226)

[2.5、提现订单审核 10](#_Toc26292)

[2.6、提现订单放款 11](#_Toc8481)

[2.7、提现订单状态通知 12](#_Toc2492)

[2.8、支付平台提现放款 12](#_Toc31545)

[2.9、订单查询 13](#_Toc3074)

[2.10、配置修改重载 13](#_Toc13401)

[三、 报表管理API 14](#_Toc19441)

[3.1、游戏信息配置 14](#_Toc12606)

[3.2、支付SDK信息配置 14](#_Toc9035)

[3.3、支付渠道配置 14](#_Toc13495)

[3.5、人工客服信息配置 14](#_Toc8555)

[3.5、其他 14](#_Toc12627)

[四、 第三方支付平台STUB 14](#_Toc5311)

[4.1、支付请求 14](#_Toc12667)

[4.2、前端真实支付 15](#_Toc17656)

[4.3、提现 15](#_Toc20833)

[4.4、echo 15](#_Toc3000)

[五、 游服STUB 16](#_Toc24942)

[5.1、支付结果回调 16](#_Toc10191)

[5.2、提现结果回调 16](#_Toc15633)

[5.3、echo 16](#_Toc18248)

[六、 安装部署 17](#_Toc289)

[6.1、支付中心 17](#_Toc26311)

[6.2、支付中心报表管理插件 20](#_Toc26160)

[6.3、支付结果回调插件 23](#_Toc22804)

[七、 报表使用说明 24](#_Toc26233)

[7.1、游戏服务器配置 24](#_Toc8135)

[7.2、后台列表配置 25](#_Toc21592)

[7.3、支付渠道配置 26](#_Toc18022)

[7.4、SDK配置 28](#_Toc28011)

[7.5、VIP客服配置 32](#_Toc15230)

[7.6、提现配置 32](#_Toc6093)

[八、 开发者指南 32](#_Toc16253)

## 概述

### 1.1、专用名词

1. 支付中心
2. 支付平台
3. 游戏服务器
4. 报表管理页面
5. URI
6. URL
7. 系统管理员
8. 运维人员
9. 平台运维人员

### 1.2、签名规则

#### 1.2.1 支付中心签名规则

1. 【json\_str】需要发送的json数据
2. 【str1】对json\_str中的**全部数据**按字典顺序，以key=value的方式，用’&’连接在一起
3. 【str2】在str1尾部增加&merchanKey=<gameCode提供的通讯密钥>
4. 【md5\_str】对str2进行MD5加密，得到**32位大写**字符串
5. 【signed\_json】在json\_str中增加一行： “sign”:”md5\_str”

signed\_json 就是向支付中心发送的数据

#### 1.2.2 游服签名规则

1. 【json\_str】需要发送的json数据
2. 【str1】对json\_str中的**全部数据**按字典顺序，以key=value的方式，用’&’连接在一起
3. 【str2】在str1尾部增加&key=<gameCode提供的通讯密钥>
4. 【md5\_str】对str2进行MD5加密，得到**32位小写**字符串
5. 【signed\_json】在json\_str中增加一行： “sign”:”md5\_str”

signed\_json 就是要向游服发送数据

### 1.3、

## 支付API

### 2.1、查询可用的支付渠道

URI: api/getChannel

提供方：支付中心

数据格式：JSON串

请求内容

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| gameCode | string | 游戏代码 |
| uin | string | 用户唯一号 |
| clientType | string | 前端类型 |
| version | Int32 | 前端版本号(内部版本号) ，-1表示不限制 |
| channelCode | string | 前端的渠道号，空表示不限制 |
| vipLevel | Int32 | VIP等级，-1表示不限制，999表示黑金账号 |
| category | string | 查询指定分类的计费点，空则表示查全部 |
| ip | String | 前端IP |
| sign | string | 签名 |

返回结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | | | | 类型 | 说明 |
| code | | | | int | 结果说明，0表示成功，其他值表示失败 |
| message | | | | string | 错误结果说明 |
| gameCode | | | | string | 游戏代码 |
| exchangeRate | | | | int | 法币与游戏金币的兑换比例，即1法币兑多少金币 |
| uin | | | | string | 用户唯一号 |
| channel[] | configId | | | int | 支付渠道ID |
| sortOrder | | | int | 排序号 |
| nameId | | | int | 支付渠道说明，用于多语言 |
| payStyle | | | string | 支付渠道代码 |
| billPointList[] | bpId | | int | 计费点ID |
| bpCode | | string | 计费点编码 |
| billPoint | | Int64 | 计费点，法币 |
| coin | | Int64 | 兑换的游戏币 |
| extraCoin | | Int64 | 赠送的游戏币 |
| initCount | | int | 可用次数，-1表示无限 |
| usedCount | | int | 已使用次数 |
| bpIcon | | string | 计费点图标 |
| bpDescId | | int | 计费点说明，用于多语言 |
| items | | string | 物品列表，以”物品ID:数量;…”表示 |
| discount | | int | 折扣，以百分比显示，仅用于显示 |
| pos | | int | 计费点序号，有可能会重复，前端自行排序 |
| extraInfo | | string | 扩展信息，以json格式表过，用于各游戏自行扩展 |
| limit | channelRange | string | 支持的渠道列表，支持正则表过式，以‘，’分隔，空表示不限制 |
| versionRange | string | 支持的版本号列表，类似于”100007500-100007700,10007900”，空表示不限制 |
| vipRange | string | 支持的VIP列表，类似于”1-3,99”，99表示为黑金用户，空表示不限制 |
| dailyCount | Int | 每日可购买数量，-1表示不限 |
| monthlyCount | int | 每月可购买数量，-1表示不限 |
| totalCount | int | 总可购买数量，-1表示不限 |
| moneyBegin | | | Int64 | 起始地法币额度 |
| moneyEnd | | | Int64 | 迄止的法币额度 |
| manualMoney | | | int | 是否支持人工输入金额，即可以moneyBegin与moneyEnd之间任意输入法币值 |
| flag | | | int | 标签：1官方推荐，2-方便快捷，3-最新 |
| iconUrl | | | string | 图标URL |
| vip[] | vipId | | | int | VIP 唯一号 |
| nameId | | | int | 名称ID，多语言用 |
| descId | | | int | 描述ID，多语言用 |
| iconUrl | | | string | 图标URL |
| flag | | | int | VIP标识，参见dict\_vipFlag表 |
| enumFlag | | | int | 内嵌标识，0-外置，1-内嵌 |
| payFlag | | | string | 支付的方式，多个支付方式以,分隔，参见dict\_vipPayFlag |
| vipUrl | | | string | 客户的链接URL |
|  |  |  | |  |  |

### 2.2、发起支付

URI: api/pay

提供方：支付中心

数据格式：JSON串

请求内容

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| configId | int | 支付渠道ID，t\_channel表中的唯一ID，代表支付方式 |
| bpId | int | 计费点ID，t\_billpoint表中的唯一ID，必填 |
| gameCode | string | 游服代码 |
| uin | string | 用户唯一号 |
| billNo | string | 游服生成的订单号 |
| amount | Int64 | 计费金额，以分计算 |
| productId | string | 产品ID，游服使用 |
| playerIp | string | 客户端用户IP |
| clientType | string | 密户端的终端类型 |
| userName | string | 玩家呢称 |
| channelId | int | 玩家渠道ID，用于分析各渠道用户付费情况 |
| channelCode | string | 玩家渠道代码 |
| channelName | string | 玩家渠道名称 |
| createTime | Int64 | 玩家创建时间，用于分析新用户付费情况 |
| sign | string | 签名 |

返回结果

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | | 类型 | 说明 |
| code | | int | 返回码，0表示成功，其他值表示失败 |
| message | | string | 错误说明，code不为0时有意义 |
| data | tradeNo | string | 支付中心生成的订单号 |
| sdk | string | 调用的SDK代码 |
| channel | string | 调用的SDK子渠道代码 |
| method | string | 支付方式代码，或子渠道的银行代码 |
| methodType | int | 支付方式类型：1-原生SDK，2-H5 |
| url | string | 前端调用url，或传给SDK的数据串（依据methodType定） |

### 2.3、支付回调

#### 2.3.1 google支付回调

【URI】: google/callback

【提供方】：支付中心

【发起方】：前端 => 游服后端 => 支付中心

【数据格式】：JSON

【请求内容】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| gameCode | string | 游戏代码 |
| uin | string | 用户唯一号 |
| orderNo | string | 支付中心的订单号 |
| productId | string | 谷歌支付的计费点代码 |
| fee | Int64 | 用户支付的金额，以分计，一般由游戏生成 |
| token | string | 谷歌/苹果支付的token串，用于校验是否合法 |
| sign | string | 签名，参见[1.2.1](#_1.2.1 支付中心签名规则) |

【返回结果】

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | | 类型 | 说明 |
| code | | int | 返回码，0表示成功，其他值表示失败 |
| message | | string | 错误说明 |
| data | result | Int | 结果说明：  1-成功，2-支付失败，3-订单取消，10-订单待确认;  400-前端数据错误,500-服务端数据错误 |
| uin | string | 用户唯一ID |
| orderNo | string | 支付中心生成的订单号 |
| googleOrderNo | string | 谷歌支付生成的订单号 |

#### 2.3.2 apple支付回调

【URI】: apple/callback

【提供方】：支付中心

【发起方】：前端 => 游服后端 => 支付中心

【数据格式】：JSON

【请求内容】

【返回结果】

#### 2.3.3 其他支付平台回调

【UR】I：依各不同平台而定

【提供方】：支付中心

【发起方】：第三方支付平台

【数据格式】：JSON

【请求内容】

不同支付平台的回调数据均不一样，不过必须包包含有支付中心生成的订单号

通过该订单号从redis缓存，或者mysql数据库中还原出订单的原始数据。

具体的操作依据支付平台的文档来确定。

【返回结果】

依据第三方支付平台的文档来确认所需要返回的结果

#### 2.3.4 通知游服支付结果

【UR】I: 由游服提供

【提供方】：游戏服务器

【发起方】：支付中心

【数据格式】：JSON

【请求内容】：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 必须 | 说明 |
| code | int | Y | 操作结果:0-成功，其他值表示失败 |
| message | string | N | 订单结果说明 |
| status | int | Y | 订单状态：  1-成功，2-支付失败，3-订单取消，10-订单待确认 |
| gameCode | string | Y | 游戏代码 |
| uin | string | Y | 用户唯一号 |
| orderNo | string | Y | 支付中心生成的订单号 |
| billNo | string | Y | 游服服务器生成的订单号 |
| productId | string | Y | 计费点代码 |
| fee | Int64 | N | 用户发起支付的金额，以分计算 |
| realFee | Int64 | Y | 实际支付的金额，以分计算 |
| coin | Int64 | Y | 计费点对应的金币数量 |
| extraCoin | Int64 | Y | 计费点对应的额外金币数量 |
| itemInfo | string | Y | 计费点对应的物品信息列表，允许多个物品：  itemId1:count1;itemId2:count2;...  itemId由游服来指定，多个物品间以’;’分隔 |
| *payId* | *Int* | *N* | *支付中心t\_payorder表中的唯一ID* |
| *channelId* | *int* | *N* | *本次支付的configId，对应t\_channel表的configId* |
| *billPoint* | *int* | *N* | *计费点ID，对应t\_billpoint表的bpId* |
| *method* | *string* | *N* | *支付方式说明，对应t\_channel表的payStyleCode* |
| category | string | N | 计费点分类信息 |
| extraInfo | string | N | 扩展信息，以json格式表达 |
| sign | string | Y | 签名，规则见 [1.2.2](#_1.2.2 游服签名规则) |

【返回结果】：

以 Content-Type: text/plain 格式返回处理结果

处理成功返回 SUCCESS, 其他内容一律认为失败

### 2.4、发起提现

【UR】I: withdraw/request

【提供方】：支付中心

【发起方】：游戏客户端

【数据格式】：JSON串

【请求内容】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| gameCode | string | 游戏代码 |
| uin | string | 用户唯一号 |
| bankCode | string | 银行代码，可以为空 |
| bankName | string | 银行账号的姓名，可以为空 |
| bankcard | string | 银行账号，必填 |
| fee | Int64 | 实际提现到账金额（已扣税），以分计算 |
| tax | Int64 | 税收金额，以分计算 |
| ip | string | 玩家IP |
| createTime | Int64 | 订单创建时间 |
| gameOrderNo | string | 游戏订单号码 |
| sign | string | 签名 |

【返回结果】

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | | 类型 | 说明 |
| code | | int | 返回码，0表示成功，其他值表示失败 |
| message | | string | 错误说明 |
| data | uin | string | 用户唯一ID |
| fee | Int64 | 发起提现的金额，以分计算 |
| orderNo | string | 支付中心生成的订单号 |

### 2.5、提现订单审核

【URI】withdraw/audit

【提供方】：支付中心

【发起方】：后台报表管理

【数据格式】：JSON串

【请求内容】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| gameCode | string | 游戏代码 |
| uin | string | 用户唯一号 |
| orderNo | string | 支付中心生成的提现订单号 |
| fee | Int64 | 提现金额，以分计算 |
| result | int | 订单审核结果   |  |  |  | | --- | --- | --- | | result | 转gamesvr | 备注 | | 1 | 0 | 待审核 | | 2 | 9 | 审核拒绝 | | 3 | 8 | 审核拒绝并金额冻结 | | 4 | 0 | 审核成功 | |
| remark | string | 审核说明 |
| operator | string | 审核人员代码 |
| auditTime | Int64 | 审核时间 |
| sign | string | 签名，规则见 1.2.1 |

【返回结果】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| code | int | 返回码，0表示成功，其他值表示失败 |
| message | string | 错误说明 |

订单状态改变后，即status字段中值对应gamesvr的值>0时须通知游服处理，协议见 [2.7](#_2.7、提现订单状态通知)。

完整版本的订单状态对照关系如下

|  |  |  |  |
| --- | --- | --- | --- |
| result | 转gamesvr | 需gamesvr处理 | 备注 |
| 1 | 0 | N | 待审核 |
| 2 | 9 | Y | 审核拒绝 |
| 3 | 8 | Y | 审核拒绝并金额冻结 |
| 4 | 0 | N | 审核成功 |
| 5 | 0 | N | 放款审核成功-SDK放款 |
| 6 | 1 | Y | 放款审核成功-人工放款 |
| 7 | 9 | Y | 放款审核拒绝 |
| 8 | 8 | Y | 放款审核拒绝并冻结资金 |
| 9 | -1 | N | 错误的数值 |
| 10 | 2 | Y | 放款成功（SDK放款） |
| 11 | 9 | Y | 放款被驳回（SDK放款） |
| 12 | 8 | Y | 放款失败（SDK放款） |
| 13 | 9 | Y | 放款订单异常（SDK放款） |

### 2.6、提现订单放款

【URI】withdraw/credit

【提供方】：支付中心

【发起方】：后台报表管理

【数据格式】：JSON串

【请求内容】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| gameCode | string | 游戏代码 |
| uin | string | 用户唯一号 |
| orderNo | string | 支付中心生成的提现订单号 |
| fee | Int64 | 提现金额，以分计算 |
| result | int | 订单发货确认结果   |  |  |  | | --- | --- | --- | | result | 转gamesvr | 备注 | | 5 | 0 | 放款审核成功-SDK放款 | | 6 | 1 | 放款审核成功-人工放款 | | 7 | 9 | 放款审核拒绝 | | 8 | 8 | 放款审核拒绝并冻结资金 | | 9 | -1 | 错误的数值 | |
| sdkCode | string | 放款所用的SDK代码,仅在result=5时有效 |
| remark | string | 放款备注 |
| operator | string | 操作人员 |
| creditTime | Int64 | 放款时间 |
| sign | string | 签名 |

【返回结果】

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | | 类型 | 说明 |
| code | | int | 返回码，0表示成功，其他值表示失败  -2: 自动放款未指定URI |
| message | | string | 错误说明 |
| data | gameCode | string | 游服代码 |
| uin | string | 用户唯一号 |
| orderNo | string | 支付中心生成的提现订单号 |
| bankCode | string | 银行代码 |
| bankCard | string | 银行账号 |
| bankName | string | 银行账户姓名 |
| fee | Int64 | 提现金额，以分显示 |
| creditRemark | string | 放款备注 |
| creditTime | Int64 | 放款时间 |
| operator | string | 放款操作人员 |
| uri | string | 自动放款的URI |
| token | string | 系统生成的交易token，平台SDK提现时需要校验 |

仅在result=5时，返回结果中才有data字段

前端拿到结果后，可以用uri来组建支付平台提现放款的请求URL

### 2.7、提现订单状态通知

【URI】：由游戏服务器来指定，可在管理页面中配置

【提供方】：游戏服务器

【发起方】：支付中心

【数据格式】：JSON串

【请求内容】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| gameCode | string | 游戏代码 |
| uin | string | 用户唯一号 |
| orderNo | string | 支付中心生成的提现订单号 |
| gameOrderNo | string | 游戏中心生成的提现订单号 |
| fee | Int64 | 提现金额，以分计算 |
| step | int | 发生的步骤:1-审核阶阶，2-放款阶段 |
| result | int | 提现订单结果：   |  |  | | --- | --- | | result | 结果说明 | | 1 | 通过，人工放款 | | 2 | 通过，SDK自动放款 | | 8 | 冻结，金额不退 | | 9 | 拒绝。金额原路返回 | |
| sign | string | 签名，规则见 [1.2.2](#_1.2.2 游服签名规则) |

【返回结果】

<TODO>

### 2.8、支付平台提现放款

【URI】依不同支付平台来确定不同的URI

【提供方】：支付中心

【发起方】：后台报表管理

【数据格式】：JSON串

【请求方式】：GET

【请求内容】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| gameCode | string | 游服代码 |
| uin | string | 用户唯一号 |
| orderNo | string | 支付中心生成的提现订单号 |
| *~~bankCode~~* | *~~string~~* | *~~银行代码~~* |
| *~~bankCard~~* | *~~string~~* | *~~银行账号~~* |
| *~~bankName~~* | *~~string~~* | *~~银行账户姓名~~* |
| *~~fee~~* | *~~Int64~~* | *~~提现金额，以分显示~~* |
| creditRemark | string | 放款备注 |
| creditTime | Int64 | 放款时间 |
| operator | string | 放款操作人员 |
| token | string | 放款确认的token |

支付中心收到数据后，需要依不同支付平台的文档，来向支付平台发起提现申请。

需要注意的是，提现申请后，支付平台会异步向支付中心发起操作结果，所以，必须依据文档的说明来操作提现回调结果，一般有两种处理方式：

1. 定时查询该提现订单的结果
2. 接收支付平台的提现结果回调

【返回结果】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| code | int | 返回码，0表示成功，其他值表示失败 |
| message | string | 错误说明 |

### 2.9、订单查询

<TODO>

### 2.10、配置修改重载

【URI】api/reload

【提供方】：支付中心

【发起方】：后台报表管理

【数据格式】：JSON串

【请求内容】

无

【返回结果】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| code | int | 返回码，0表示成功，其他值表示失败 |
| message | string | 错误说明 |

### 2.11 订单取消

URI: api/paycancel

提供方：支付中心

数据格式：JSON串

请求内容

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| gameCode | string | 游服代码 |
| uin | string | 用户唯一号 |
| billNo | string | 游服生成的订单号 |
| ip | string | 前端IP |
| sign | string | 签名 |

### 2.11、订单状态字典

## 报表管理API

### 3.1、游戏信息配置

### 3.2、支付SDK信息配置

### 3.3、支付渠道配置

### 3.5、人工客服信息配置

### 3.5、其他

## 第三方支付平台STUB

这个工具是用于模拟第三方支付平台在支付中心的响应，主要是为了测试方便。

该模块位于 webmw/stub\_paysdk 目录下

### 4.1、支付请求

【URI】：paystub/payreq

【提供方】：stub\_paysdk

【发起方】：支付中心

【数据格式】：JSON串

【请求内容】

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| merchant | string | 商户代码 |
| uin | string | 用户唯一码 |
| userIp | string | 用户的IP |
| productId | string | 订单ID |
| payStyle | string | 支付方式，一般用银行码 |
| money | Int64 | 支付金额 |
| tradeNo | string | 支付中心生成的交易订单号 |
| notify\_url | string | 回调通知URL |

【返回结果】

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | | 类型 | 说明 |
| code | | int | 返回码，0表示成功，其他值表示失败 |
| message | | string | 错误说明 |
| data | tradeNo | string | 支付中心生成的订单号 |
| orderNo | string | 支付平台生成的订单号 |
| money | Int64 | 交易金额，以分计算 |
| url | string | 真实支付的URL  <http://ip:port/uri?uin=xxx&money=xxx&tradeNo=xxx&orderNo=xxx>  Ip,port,uri 一般配置在config.xml中 |

### 4.2、前端真实支付

【URI】：paystub/pay

【提供方】：stub\_paysdk

【发起方】：支付中心

【数据格式】：JSON串

【请求内容】

【返回结果】

### 4.3、提现

【URI】：paystub/withdraw

【提供方】：stub\_paysdk

【发起方】：支付中心

【数据格式】：JSON串

【请求内容】

【返回结果】

### 4.4、echo

【URI】：paystub/echo

【提供方】：stub\_paysdk

【发起方】：支付中心

【数据格式】：JSON串

【请求内容】

【返回结果】

## 游服STUB

这个工具是用于模拟游戏服务器在支付中心的响应，主要是为了测试方便

该模块位于 webmw/stub\_gamesvr 目录下

### 5.1、支付结果回调

### 5.2、提现结果回调

### 5.3、echo

## 安装部署

### 6.1、支付中心

支付中心主程序是payframe，每个支付接口为独立的.so，配置在config.xml中的plugin节中。

|  |  |  |
| --- | --- | --- |
| Config.xml |  | 参数配置说明 |
| <? xml version="1.0" encoding="utf8" ?> |  |  |
| <main> |  |  |
| <log level="3" module="payv3" path="/data/work/log" /> |  |  |
| <service pid="./paycenter.pid" type="http" header="1" > |  |  |
| <net id="1" ports="3310" ssl="0" /> |  |  |
| <accept timeout="100" parallel="10" /> |  |  |
| <worker count="1" startId="1000" > |  |  |
| <client start="10000" maxconn="100000" step="1000" /> |  |  |
| </worker> |  |  |
| <client> |  |  |
| <active timeout="500" /> |  |  |
| <authorize timeout="500" /> |  |  |
| <heartbeat interval="5000" timeout="3" /> |  |  |
| <resp buffer\_size="65536" /> |  | 返回包文缓冲大小，若超过该值则启用压缩 |
| </client> |  |  |
| <alert ids="88000015,88100003" url="http://webmw\_hk.opk2.com:18081/bot/sendMsg" note="充值测试 中心"/> |  | 出错时的告警数据 |
| </service> |  |  |
| <plugin empty\_cgi="index.html" > |  | 插件管理 |
| <cgi name="测试STUB" slot1="./webpay/webpay\_dummy.so" slot2="" init="doInit" entry="queryRegister" maintance="run" /> |  |  |
| <cgi name="提现管理" slot1="./webpay/webpay\_withdraw.so" slot2="" init="doInit" entry="queryRegister" maintance="run" /> |  |  |
| </plugin> |  |  |
| <dbms> |  | 数据库连接配置 |
| <db id="1" name="redis" type="redis" host="127.0.0.1" port="6379" db="2" user="" passwd="" /> |  |  |
| <db id="2" name="paycenter" type="mysql" host="127.0.0.1" port="3306" db="paycenter\_v3" user="payuser" passwd="cXc0OTF5cThlMTc3NmI1ODBiMGQ=" /> |  |  |
| </dbms> |  |  |
| <clientIp source="header" prop="X-Real-IP" /> |  | 玩家IP获得方式 source = [net|header|param]  net 是指从IClient\* 获取， 如 pClient->getIpStr()，忽略 prop  header 是指从 CHttpRequest 的 header中获取, prop是指名称  param 是指从 CHttpRequest 的 param 中获取，prop是指名称 |
| <authorize active="on"> |  | 是否启用API认证，active=[on|off]  source = [header|param|cookie]  user = 用户名字段  sign = 密钥字段 |
| <field source="param" user="user" sign="sign" /> |  |  |
| </authorize> |  |  |
| <paycenter sign="on" mgr\_sign="on"> |  | 支付中心主要的配置 |
| <db main="paycenter" redis="redis" /> |  | 支付中心涉及的数据库以及redis配置，名称从dbms节获得 |
| <uri\_pay uri="api/pay"/> |  | 支付的CGI |
| <uri\_reload uri="api/reload" validIp="127.0.0.1" /> |  | 重载配置的CGI，以及允许调用该命令的IP列表，IP可以多个，以','分隔 |
| <uri\_getchannel uri="api/getChannel" /> |  | 获得可用支付渠道的CGI |
| <apply url="http://127.0.0.1:3310/api/reload" /> |  |  |
| <timezone adjust="-3600" /> |  | 时区校正值，用于跨区部署时用 |
| <ip enable="true" > |  |  |
| <report valid="0.0.0.0" desc="报表后台" /> |  |  |
| <gamesvr valid="0.0.0.0" desc="游戏服务器" /> |  |  |
| </ip> |  |  |
| <sign> |  |  |
| <recharge enable="on" field="sign" desc="充值请求是否需要签名" /> |  |  |
| <withdraw enable="on" field="sign" desc="代付请求是否需要签名" /> |  |  |
| <report enable="on" field="sign" desc="报表管理系统是否要签名" /> |  |  |
| </sign> |  |  |
| </paycenter> |  |  |
| <dummy sdk="DUMMY" desc="测试支付" /> |  | 支付中心的插件配置，用于指定SDK代码，各插件可定制自己的配置 |
| <pay77 sdk="PAY77" desc="越南-PAY77" withdraw\_ip="0.0.0.0" > |  |  |
| <ip enable="true"> |  |  |
| <recharge valid="0.0.0.0" desc="充值回调IP, 多个IP以','分隔"/> |  |  |
| <withdraw valid="0.0.0.0" desc="代付回调IP, 多个IP以','分隔"/> |  |  |
| </ip> |  |  |
| </pay77> |  |  |
| </main> |  |  |

### 6.2、支付中心报表管理插件

报表中心主程序是webframe，必须的插件有plugin\_paycenter.so、plugin\_rpt\_mysql.so、plugin\_system.so、plugin\_ipdata.so 四个

|  |  |  |
| --- | --- | --- |
| Config.xml配置 |  | 参数说明 |
| <plugin empty\_cgi="ip/ipinfo" > |  |  |
| <cgi name="paycenter" slot1="./plugins/plugin\_paycenter.so" slot2="" init="doInit" entry="queryRegister" maintance="run" /> |  | 支付中心配置管理的API |
| <cgi name="report" slot1="./plugins/plugin\_rpt\_mysql.so" slot2="" init="doInit" entry="queryRegister" maintance="run" /> |  | 支付中心报表查询API入口，对应配置 <report> |
| <cgi name="system" slot1="./plugins/plugin\_system.so" slot2="" init="doInit" entry="queryRegister" maintance="run" /> |  | 提供一些上传下载服务的API |
| <cgi name="ipdata" slot1="./plugins/plugin\_ipdata.so" slot2="" init="doInit" entry="queryRegister" maintance="run" /> |  | 提供IP查询的API，对应配置<ipdata> |
| </plugin> |  |  |
| <ipdata datfile="./ip.dat" /> |  | IP文件 |
| <report format="./report.xml" dbms="paycenter,dt\_hall" api="api/report/.\*" /> |  | 报表配置文件名，以及对应的API 别名,api支持正则表达式配置 |
| <paycenter sign="false" mgr\_sign="false" > |  | 支付中心配置管理的配置 |
| <db main="paycenter" mysql="paycenter" hall\_mysql="dt\_hall" redis="common\_redis" /> |  |  |
| <apply url="http://127.0.0.1:3310/api/reload" /> |  |  |
| </paycenter> |  |  |

report.xml 样板

|  |  |  |
| --- | --- | --- |
| Report.xml |  | 参数说明 |
| <? xml version="1.0" encoding="utf-8" ?> |  |  |
| <api> |  |  |
| <report> |  | 报表入口 |
| <item name="prc\_get\_rolemanager\_channelall" comment="查询渠道的角色数据"> |  |  |
| <procedure data="prc\_get\_rolemanager\_channelall" count="" /> |  | data=> 查询数据的存储过程名称 count=> 查询数据总量的在储过程名称，允许为空 |
| <param source="param" name="p\_roleid" param="0" type="C" flag="1" empty\_as\_null="false" /> |  | <param>参数：   |  |  | | --- | --- | | source=param | 从 http 的 param 中获得 | | source=header | 从 http的 header中获得 | | name | header/param 中的名字 | | param | 存储过程的参数名称或序号，mysql中以序号为准 | | type | param 的类型: C-字符串，I-整理, F-浮点数, [缺省为'C', 可不填] | | flag | 参数标识: 1-只用在data过程中; 2-只用在count过程中; 3-两个过程都使用；[缺省为3, 可不填] | | empty\_as\_null | name无数据时，param是否设为NULL, [缺省为 false 可以不填] | |
| <param source="header" name="manageKey" param="1" type="C" flag="1" empty\_as\_null="false" /> |  |
| </item> |  |
| <item name="GetSDK" comment="SDK查询" > |  |
| <procedure data="pGetSDK" count="pGetSDKCNT" dbms="paycenter" /> |  |
| <param source="param" name="p\_gameCode" param="0" type="C" /> |  |
| <param source="param" name="p\_record\_start" param="1" type="I" flag="1" /> |  |
| <param source="param" name="p\_record\_num" param="2" type="I" flag="1" /> |  |
| </item> |  |
| </report> |  |  |
| </api> |  |  |

### 6.3、支付结果回调插件

## 报表使用说明

### 7.1、游戏服务器配置

所在位置：支付后台=>游戏支付管理

操作人员：运维管理人员、系统管理员



1. 游戏编码：关键字之一，在支付中心系统中保持唯一
2. 游戏名称：助记词
3. 商户ID：游戏服务器提供，通讯之用
4. 商户密匙：游戏服务器提供，用于支付中心与游服之间通讯协议的签名之用
5. 兑换比例：即法币与金币之间的兑换比例
6. WEB支付成功返回地址：支付成功后，通知客户端的调用地址，由游戏前端指定
7. 支付成功通知地址：支付成功后，通知到游服进行增加金币和物品，由游服提供
8. 下分成功游戏通知地址：提现成功后，通知游服进行扣减金币的地址，由游服提供
9. 白名单列表：支付中心认可的游戏服务器来源IP，0.0.0.0 表示不限；允许多个来源，以’;’进行分隔。一般情况下不需要设置
10. 警告数据：支付出异常时通知到IM平台，是一个JSON格式的数据，{“alertId”:8001234,”url”:”http://www.webmw.com:8081”}，其中:

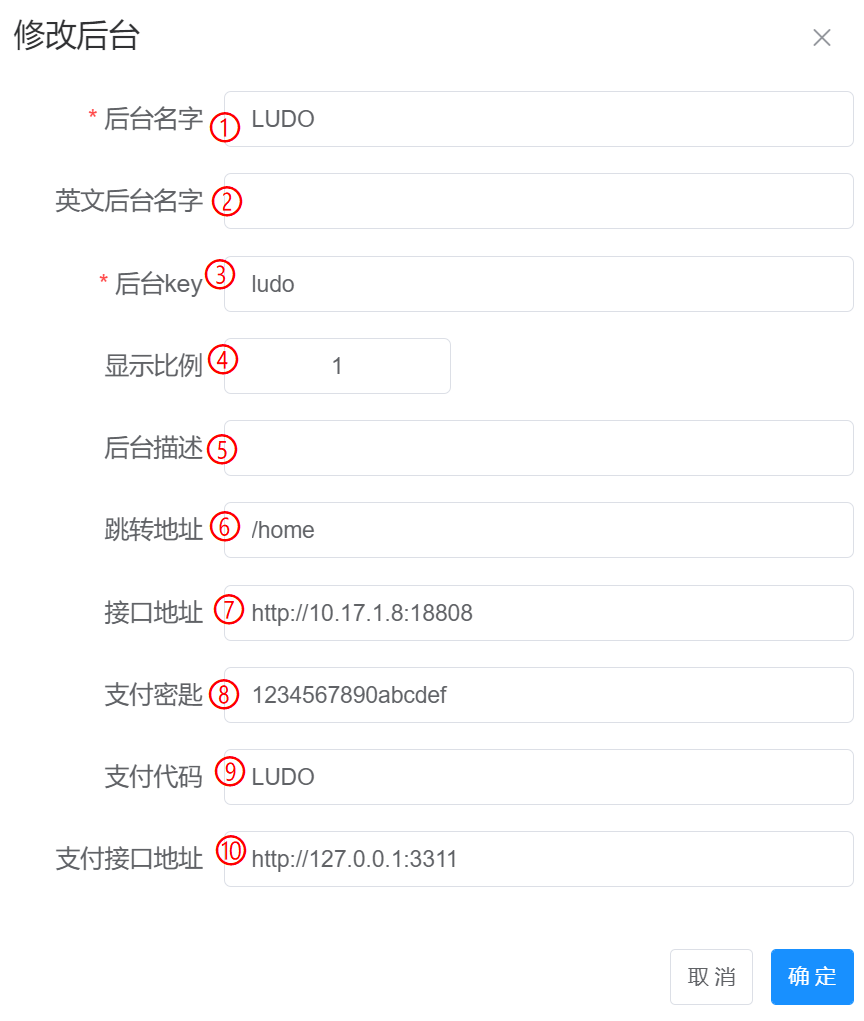
alertId:告警通知的ID，请联系运维人员获得

url: 告警通知的地址，目前我们有 webmw.opk2.com webmw-hk.opk2.com 两个地址，端口号均为 18081

### 7.2、后台列表配置

所在位置：权限管理=>后台列表

操作人员：运维人员、系统管理员



1. 后台名字：显示在主界面上的后台系统名字
2. 英文后台名字：英文版本的后台名字
3. 后台KEY：唯一标示key值，是提取后台的数据的关键值
4. 显示比例：数据金币的比列值，默认是1:10000显示
5. 后台描述：
6. 跳转地址：点击管理后台入口，默认跳转的页面地址
7. 接口地址：管理后台跟游戏的接口地址，用于游戏提供的参数配置API调用
8. 支付密匙：用于与支付中心通讯的密钥，与7.1中的商户密钥一致
9. 支付代码：用于与支付中心通讯的代码，与7.1中的商户ID一致
10. 支付接口地址：支付中心的地址

### 7.3、支付渠道配置

#### 7.3.1、支付渠道TAB页配置

所在位置：支付管理=>SDK充值配置=>支付渠道配置

操作人员：运维管理人员、系统管理员

每一个支付渠道在前端中为单独一个TAB页



1. 支付方式：选择支付方式，该数据从dict\_paystyle表中配置，对于谷歌支付要选“谷歌应用商店”，苹果支付要选“苹果应用商店”，不清楚选哪一个时，请咨询运维人员。
2. 最小金额：该支付渠道的最小支付金额
3. 最大金额：该支付渠道的最大支付金额
4. 输入金额：是否支持人工输入金额
5. 标识：在支付渠道TAB页上显示的标识信息，如“官方推荐”、“充值优惠”、“最新”等
6. 图标：在支付渠道TAB页上显示的图标，可以用URL显示，或者前端程序所支持的格式，具体需要咨询运维人员。
7. 语言标识ID:在多语言时，支付渠道TAB页上要显示的文字的ID号
8. 排序：该支付汇道的排序号，从小到大排序，序号越小越前面。

#### 7.3.2、计费点配置

所在位置：支付管理=>SDK充值配置=>支付渠道配置

操作人员：平台方运维人员



在支付渠道的“通用计费点”上可以配置计费点，选中计费点，如“test\_1800”可以直接修改计费点的配置；点击“新增”可以增加新的计费点。



1. 计费点代码：主键，在同一个支付渠道中计费点代码唯一。若是谷歌支付，则该计费点代码必须与谷歌应用商城上的计费点一致；苹果支付也一样。
2. 通用计费点：实际上是“付款金额”
3. 金币：该计费点对应的金币数量，若是物品或其他特殊道具时，该值设为0
4. 赠送金币：该计费点对应的额外赠送金币数量。
5. 购买次数上限：该计费点可购买次数，-1表示无限次，0表示不可购买
6. 计费点图标：该计费点在商城中显示的图标，一般为URL。
7. 计费点描述ID：用于描述该计费点的具体内容，用于多语言环境，指示文本的ID
8. 计费点数据项：计费点的扩展描述，用于购买钻石、VIP、道具等。格式为：物品ID[:物品数量]，其中物品数量可选，多个物品信息以’;’来分隔。

计费点一般不支持删除。

### 7.4、SDK配置

所在位置：支付管理=>SDK充值配置=>SDK配置

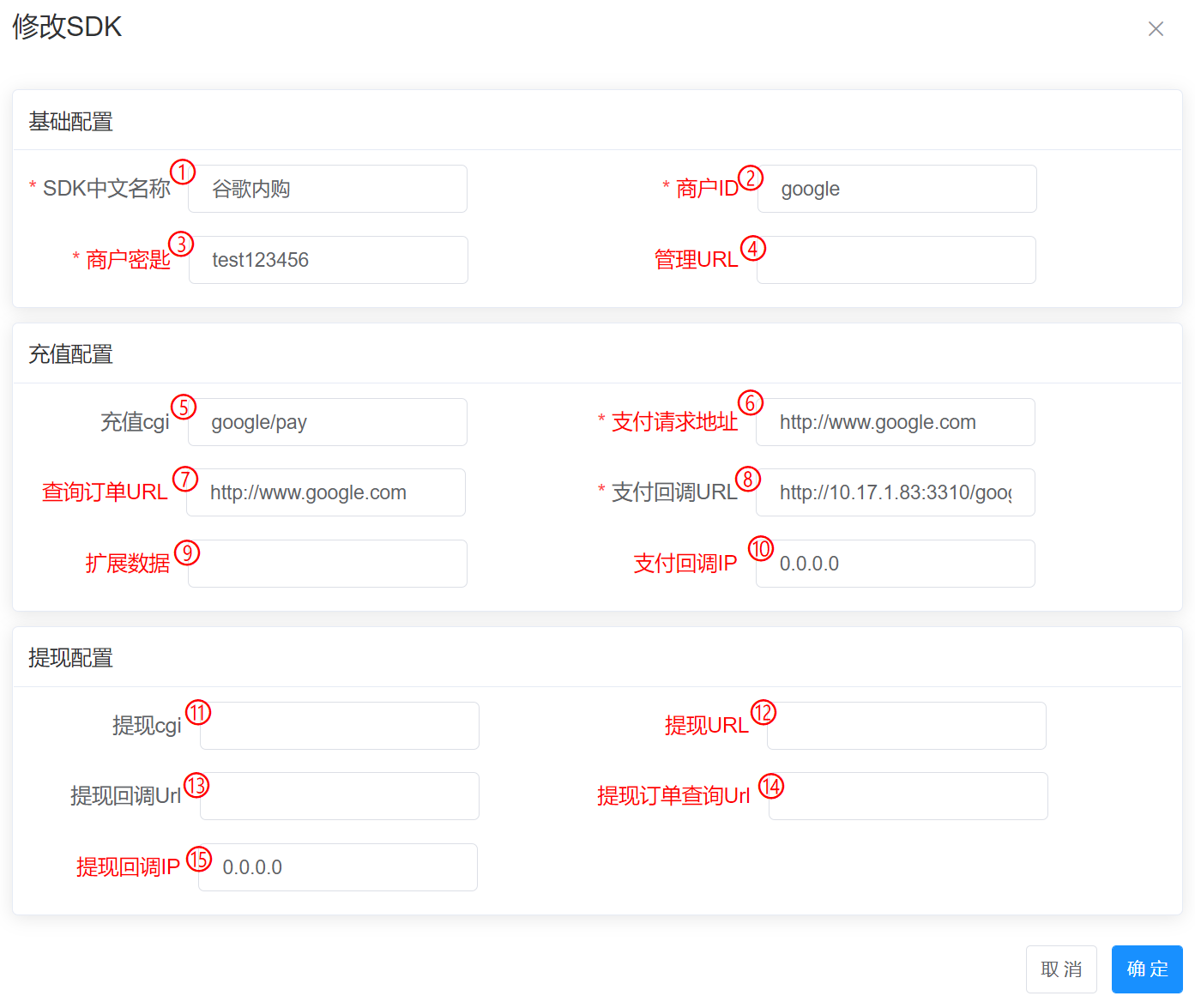
操作人员：运维人员、系统管理员



点击“配置”进入详细配置页面：

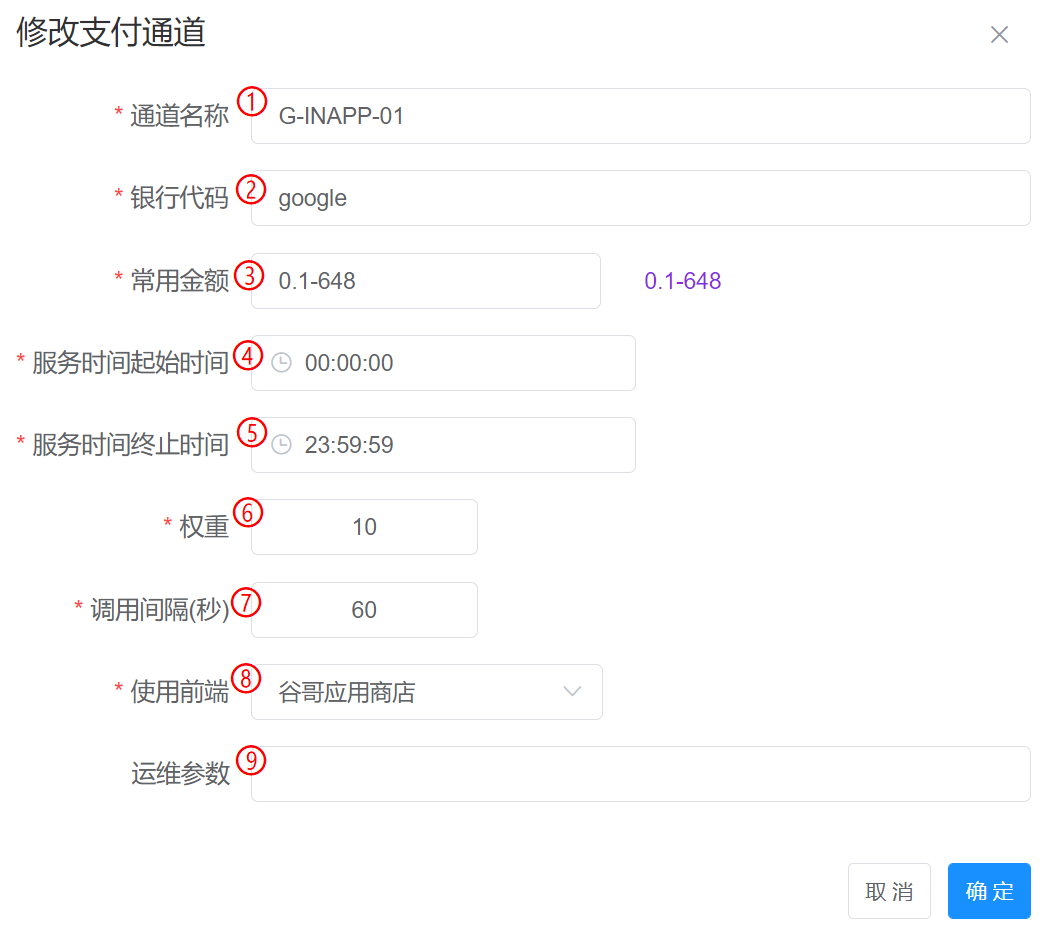


#### 7.4.1、基础信息配置



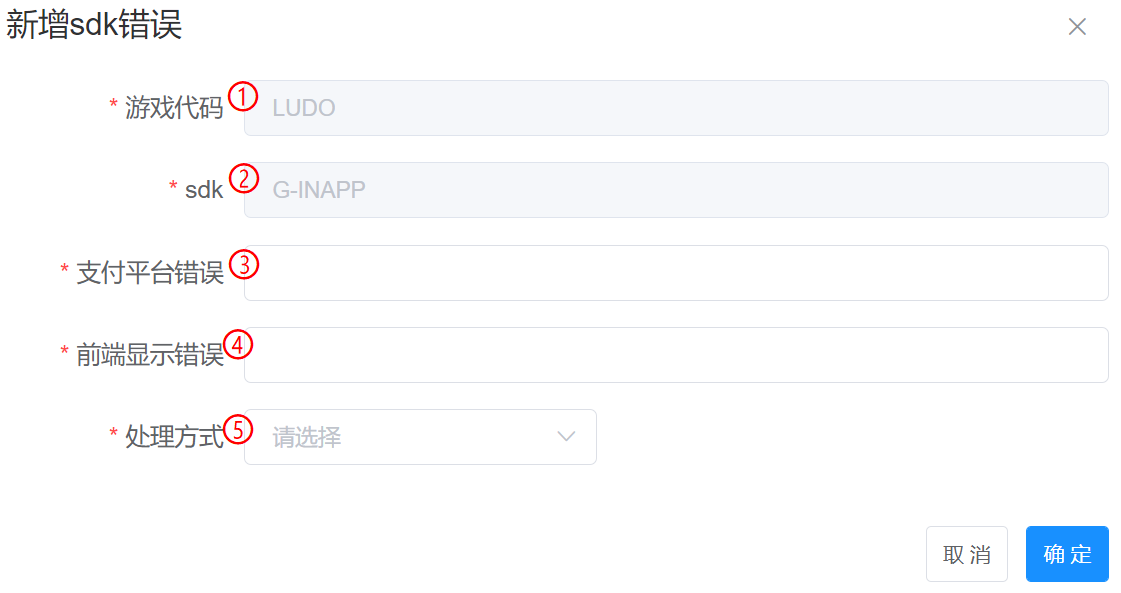
1. SDK中文名称：配置的SDK助记词
2. 商户ID: 第三方支付平台提供的商户ID
3. 商户密匙：第三方支付平台提供的通讯密钥
4. 管理URL：第三方支付平台的URL，方面从支付中心直接进入管理页面。
5. 充值CGI：支付中心提供的URI，用于支付插件的实际调用。此URI不论是否有通过nginx中转，都不需要加额外的数据
6. 支付请求地址：第三方支付平台的充值请求URL
7. 查询订单URL：第三方支付平台的充值订单查询URL
8. 支付回调URL：支付中心提供的充值回调URL
9. 扩展数据：第三方平台支付时所需要的额外数据，需与支付中心的程序员联系获得数据。一般为JSON格式
10. 支付回调IP：第三方平台充值回调时的IP，支持多个IP，多个IP之间以’,’分隔。若不检查回调IP，则以“0.0.0.0”填充
11. 提现CGI：支付中心提供的代付URI，若有能过nginx中转时，依据nginx的配置来添加所需要的信息，如：插件提供的URI为google/withdraw，则必须在URI前面加上api/，以便nginx能正确转发。具体添加的信息由系统维护人员提供。
12. 提现URL：第三方支付平台的代付请求URL
13. 提现回调URL:支付中心提供的代付请求回调时调用URL，用于获得代付结果的通知。
14. 提现订单查询URL：第三方支付平台提供的代付订单结果查询URL
15. 提现回调IP：检查⑬项的调用时的IP地址,若不需要检查回调IP，则可以填“0.0.0.0”

#### 7.4.2、SDK通道配置



1. 通道名称：通道的助记词
2. 银行代码：该SDK子通道的银行代码，对于谷歌支付和苹果支付，随意填
3. 常用金额：该SDK子通道支持的金额范围，支持单值和范围值，如0.1-648表示从0.1元到648元之间都支持，范围以’-’的分隔；如648则表示仅支持该值；若需要配多个金额范围，则以’;’分隔。
4. 服务起始时间：该SDK子通道启用的时间
5. 服务终止时间：该SDK子通道停用的时间，一般设 00:00:00:-23:59:59 表示全天
6. 权重：同一个支付渠道中占用的权重，权重越大则被调用的概率越高。该值对于同一个支付渠道中多个SDK子通道时有意义。
7. 调用间隔(秒)：暂未用
8. 使用前端：指定该SDK子通道应用于哪个环境，如“通用”表示可以应用在除谷歌支付、苹果支付外的全部环境；“安卓”表示仅用于安卓手机；“苹果”表示仅用于苹果手机；“谷歌应用商店”表示仅支持谷歌支付；“苹果应用商店”表示仅支持苹果支付。
9. 运维参数：一般以JSON来表达，具体由支付插件的开发人员来提供

#### 7.4.3、SDK错误配置



1. 游戏代码：<不可更改>
2. SDK：<不可更改>
3. 支付平台错误：支付平台返回的错误内容
4. 前端显示错误：显示给前端的错误描述，支持多语言ID
5. 处理方式：SDK出现错误信息时对应的处理方式：停用计费点、停用SDK等

### 7.5、VIP客服配置

### 7.6、提现配置

## 开发者指南