

```
import Callout from 'next-theme-docs/callout'; import PaymentSequenceDiagram from './assets/new-payment-sequence-diagram.svg';
```

支付接口

时序图

流程

- 1. 登录Publisher 工具获得支付 `clientId` 与 `clientSecret`, 并设置接收支付 IPN 通知 URL
- 2. 创建全局认证: 通过 (POST: `/api/cp/token`) 向平台请求全局认证 token
- 3. 创建订单: 获取平台订单号 (`orderNo`)
- 4. 启动支付窗口: 游戏前端调启平台支付窗口 (前端 API)
- 5. 玩家支付
- 6. 接收 & 处理 IPN: 游戏服务器、确认、处理 IPN 并发货
- 7. 上报发货状态到平台: 游戏服务器向平台上报发货结果

Demo 演示

环境参数

系统	环境	API Host
支付	测试	https://psp.stg.g123.jp
支付	正式	https://psp.g123.jp

从安全性上考虑，所有的平台支付 API 调用都应在游戏服务器后端上完成。

1. 创建全局认证 token

通过此 API 可以得到用于进行其他 API 调用的全局 `token`。创建的多个 token 在有效期内可同时使用，但需注意避免频繁创建新的 token。

## Endpoint

**POST** {API Host}/api/cp/token

## Request Header

Request header	Description
Content-Type	application/json

## Request Body

Request

Body	Property	Description
grant_type	client_credential	默认值不需要修改
client_id	your clientId	从 Publisher 获取
client_secret	your clientSecret	从 Publisher 获取

## Response

Response Description

access_token	Access token; 请避免频繁创建新 token ; 在失效前 5 分钟左右创建即可
token_type	Bearer
expires_in	Expire time(in seconds)
created_at	The token created time (Unix timestamp in milliseconds); 即失效时间点 = created_at + expires_in*1000

## 用例

### Request

```
curl -X POST \  
  {API Host}/api/cp/token \  
  -H 'Content-Type: application/json' \  
  -d '{ "grant_type": "client_credential", "client_id": "your clientId", "client_secret": "your clientSecret" }'
```

```
-H 'Content-Type: application/json' \  
-H 'cache-control: no-cache' \  
-d '{  
  "grant_type": "client_credentials",  
  "client_id": "testapp_cL9-ckoA2",  
  "client_secret": "Wqrl1bHFRoF07LdYIIIZ71A9pClsDlcrC"  
}'
```

## Response

```
{  
  "access_token": "K4RVtZjt1tXPSAWnADhZhZsMJ3l1fiI8",  
  "token_type": "Bearer",  
  "expires_in": 86400,  
  "created_at": 1547972209599  
}
```

## 2. 创建订单

通过此 API 在 G123 支付平台上创建订单，并获取平台返回的订单号。同一个订单号在前端只能激活一次支付窗口。

### 防止金额被篡改方法（非常重要）

#### 1. 防止金额在浏览器端被篡改：

1. （商品 Code，币种）→ 金额 的对应关系保存在游戏服务器端。
2. 前端只发送商品 Code 到游戏服务器，不可以直接发送金额。
3. 在游戏服务器上找到金额后拼装，创建订单请求后，由服务器端发送到平台支付 API。

#### 2. 防止币种被篡改：请使用通过平台账号接口获取到的币种

## Endpoint

**POST** {API Host}/api/cp/orders

## Request Header

Request header	Description
Content-Type	application/json
Authorization	Bearer \${accessToken}

## Request Body

Request Body	Description	Mandatory
version	3.3	YES
items	item array, json 结构参照下表	YES
custom	custom json, json 结构参照下表	YES

## item

Key	Value	Limitation&Description	Default	Mandatory
code	商品代码 (游戏内商品标识码)	64 byte	N/A	YES
name	商品名称	64 byte	N/A	YES
amt	商品价格	0.01 {'<='} amt {'<='} 200000	N/A	YES
currency	币种	使用平台2. 账号接口获取的币种码	N/A	YES
desc	商品简介	64 byte	Empty	NO
qty	商品数量 (虚拟电子产品必须为 1)	qty {'>='} 1	1	NO
taxamt	税额 (可以把税额含入 amt 之中, 并把此值赋 0)	taxamt {'>='} 0	0	NO
url	商品 URL	255 byte	Empty	NO

- 多币种价格为非实时汇率换算，请在此获取 平台多币种价目对照表 - 币种仅由用户所在国家/地区决定，与语言无关

### custom

Key	Value	Description	Type	Mandatory
game_server_id	游戏服务器		String	YES
item_type	道具类型 (pass/fund/item/gift) 字符	游戏内没有定义可传空	String	YES
role_id	角色 ID		String	YES
role_level	角色等级		String	YES
role_vip_level	角色 VIP 等级		String	YES
reason	充值原因, 活动等	游戏内没有定义可传空 字符	String	YES
virtual_currency	获得虚拟货币数量	游戏内没有定义可传"0"	String	YES
notify_url	订单通知的回调地址	<b>需要事先登记</b>	String	YES
ctwid	G123 ID	g123 平台发行的用户 唯一 id	String	YES

尽量上传所有字段，若无法获取某字段值则忽略，但不要上传空值。

notify\_url 需要在事先平台 Publisher 工具 (G123 接入 > 支付) 里进行配置

### Response

Response	Description
orderNo	订单号.
token	订单 Token.

### 用例

### Request

```
curl -X POST \  
  {API Host}/api/cp/orders \  
  -H 'Authorization: Bearer K4RVtZjt1tXPSAWnADhZhZsMJ3l1fiI8' \  
  -H 'Content-Type: application/json' \  
  -H 'cache-control: no-cache' \  
  -d '{  
    "version": "3.3",  
    "items": [  
      {  
        "code": "special_bag",  
        "name": "Some Item",  
        "desc": "Some desc",  
        "amt": 1050,  
        "currency": "JPY",  
        "qty": 1,  
        "taxamt": 0,  
        "url": ""  
      }  
    ],  
    "custom": {  
      "user": {  
        "id": "user234",  
        "desc": "  聖闘士星矢"  
      },  
      "game_server_id": "osaka_1",  
      "item_type": "gift",  
      "role_id": "as",  
      "role_level": "150",  
      "role_vip_level": "3",  
      "reason": "  新年活动",  
      "virtual_currency_amount": "12",  
      "notify_url": "https://www.cpgame.com/ipn/receiver", // 请事先登记  
      "ctwid": "G12345678"
```

```
    }  
  }'  

```

## Response

```
{  
  "orderNo": "0241901200002606",  
  "token": "52de3219-2efb-4a98-b175-c9b19e0be5f9"  
}
```

## 订单格式校验

平台使用 JSON Schema 对订单的格式进行校验，格式错误时会返回 `invalid_request` 错误。

可以通过在线校验工具对要发送的内容进行校验。具体方法如下

1. 打开工具: <https://www.jsonschemavalidator.net>
2. 将下面的 JSON Schema 贴入左侧的文本框
3. 将要发送的 Request Body (JSON) 内容贴入右侧的文本框
4. 校验工具的下方会显示校验结果

请拷贝此 JSON Schema, 贴入上述校验工具

```
{  
  "additionalProperties": false,  
  "type": "object",  
  "properties": {  
    "version": {  
      "type": "string",  
      "enum": ["3.3"]  
    },  
    "custom": {  
      "type": "object",  
      "additionalProperties": true,  
      "properties": {  
        "game_server_id": {  

```

```
    "type": "string",
    "minLength": 1,
    "maxLength": 64
  },
  "item_type": {
    "type": "string",
    "minLength": 1,
    "maxLength": 64
  },
  "role_id": {
    "type": "string",
    "minLength": 1,
    "maxLength": 64
  },
  "role_level": {
    "type": "string",
    "minLength": 1,
    "maxLength": 8
  },
  "role_vip_level": {
    "type": "string",
    "minLength": 1,
    "maxLength": 8
  },
  "reason": {
    "type": "string",
    "minLength": 1,
    "maxLength": 64
  },
  "virtual_currency_amount": {
    "type": "string",
    "minLength": 1,
    "maxLength": 64
  }
```



```
    },
    "notify_url": {
      "type": "string",
      "minLength": 1,
      "maxLength": 255
    },
    "ctwid": {
      "type": "string",
      "minLength": 7,
      "maxLength": 10
    }
  },
  "required": [
    "game_server_id",
    "item_type",
    "role_id",
    "role_level",
    "role_vip_level",
    "reason",
    "virtual_currency_amount",
    "notify_url",
    "ctwid"
  ]
},
"items": {
  "type": "array",
  "minItems": 1,
  "items": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
      "code": {
        "type": "string",
```

```
        "minLength": 1,
        "maxLength": 64
    },
    "name": {
        "type": "string",
        "minLength": 1,
        "maxLength": 64
    },
    "desc": {
        "type": "string",
        "maxLength": 64,
        "default": ""
    },
    "amt": {
        "type": "number",
        "minimum": 0.01,
        "maximum": 1000000
    },
    "currency": {
        "type": "string",
        "minLength": 3,
        "maxLength": 3
    },
    "qty": {
        "type": "number",
        "minimum": 1,
        "default": 1
    },
    "taxamt": {
        "type": "number",
        "minimum": 0,
        "default": 0
    },
    },
```

```
        "url": {
            "type": "string",
            "maxLength": 255,
            "default": ""
        }
    },
    "required": ["code", "name", "amt", "currency"]
}

}

},
"required": ["version", "items", "custom"]
}
```

JSON Schema Validator

An online, interactive JSON Schema validator. Supports JSON Schema Draft 3, Draft 4, Draft 6, Draft 7 and Draft 2019-09.

Select schema: Custom

Input JSON:

贴入json schema

待校验的JSON

✓ No errors found. JSON validates against the schema

图 1: json schema validation

### 3. 启动支付窗口

获取订单号 (orderNo) 后, 游戏前端通过调用 激活支付窗口 API 启动平台支付窗口。如:

// 传 *JSON object* (不是字符串), 无需序列化

```
<script>  
    window.CpSdk.sendPayment({  
        "orderNo": "0241901200002606",  
        "token": "52de3219-2efb-4a98-b175-c9b19e0be5f9"  
    });  
</script>
```

### 4. 模拟用户支付 (测试环境)

只需测任意一种支付方式:

- amazon pay:

测试账号密码

g123@gmail.com

1234zxcv

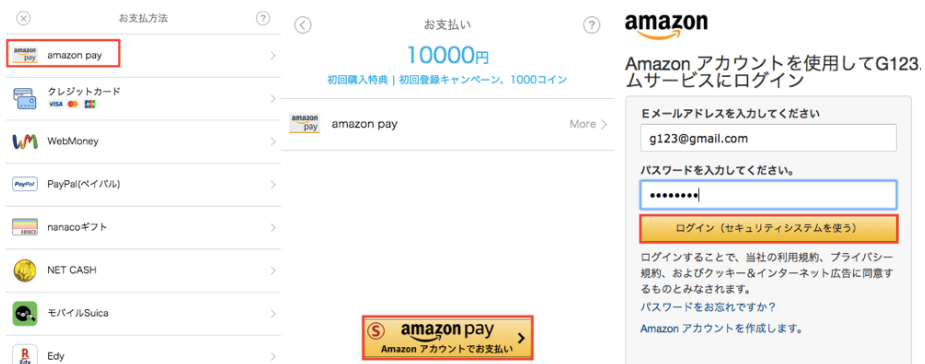


图 2: test3

- クレジットカード (信用卡):

Card Type: JCB

Card Number: 3569990010095841  
Expiration Date: 03/2030  
CVV: 737

Card Type: Visa  
Card Number: 4166676667666746  
Expiration Date: 03/2030  
CVV: 737

Card Number: 4035501000000008  
Expiration Date: 03/2030  
CVV: 737

Card Type: Mastercard  
Card Number: 2222400010000008  
Expiration Date: 03/2030  
CVV: 737

Card Number: 5555444433331111  
Expiration Date: 03/2030  
CVV: 737

Card Type: American Express(Amex)  
Card Number: 370000000000002  
Expiration Date: 03/2030  
CVV: 7373

其他 Adyen 的测试卡号: <https://docs.adyen.com/development-resources/testing/test-card-numbers/>

- PayPal:

测试账号密码  
game-cp@g123.jp  
12345678

?

お支払方法

×

クレジットカード

VISA

1.信用卡

>

WebMoney

>

NET CASH

>

モバイルSuica

>

Edy

>

nanacoギフト

>

PayPal

PayPal(ペイパル)

2.Paypal

>

クレジットカードで支払う

カード番号

3546985526870834

支払いタイプ

VISA

MasterCard

JCB

有効期限

月(mm)

03

/

年(yy)

29

セキュリティコード

428

説明を見る

今すぐ支払う

PayPal

PayPalで支払う

game-cp@g123.jp

.....

☐ ログイン状態を保持する

ログイン

PayPal

¥10,000 JPY

Hi shangin.

Choose a way to pay

☐ PayPal Balance
 ☒ VISA Visa x-6256

+ Add a card

Continue

PayPal is the safer, faster way to pay  
 No matter where you shop, we keep your financial information secure.

图 3: test2

- Paidy:

测试账号密码

电话：08000000001

验证码：8888

支付成功邮箱：successful.payment@paidy.com

支付失败邮箱：rejected.payment@paidy.com

The image shows two side-by-side screenshots of the Paidy registration interface. The left screenshot shows the registration form with the email address 'successful.payment@paidy.com' and the phone number '08000000001' entered in red-outlined boxes. The right screenshot shows the verification step with the text '携帯電話に届いた4桁の認証コードを入力してください' (Please enter the 4-digit verification code received on your mobile phone) and a red-outlined box containing the code '8888'.

G123.jp

メールアドレス

successful.payment@paidy.com

携帯電話番号

08000000001

☐ 次回から入力を省略

利用規約・個人情報取扱条項に同意して

次へ

G123.jp

080-0000-0001 successful.payment@paidy.c... 編集

携帯電話に届いた4桁の認証コードを入力してください

8 8 8 8

图 4: test3

- PayPay:

测试账号密码

电话：09025177804

密码：Y7RFsPPT82

验证码：1234 or 1111


[ログイン](#)

PayPayに登録した携帯電話番号でログイン

登録済みの携帯電話番号

090-2517-7804

パスワード

パスワードをお忘れですか？

ログイン

外部のアカウントでログイン

 Yahoo! JAPAN ID

アカウントをお持ちでない場合 [新規登録](#)

[日本語](#) [English](#)





SMSで届いた認証コードを入力してください

こちらの携帯電話番号に送信しました090-2517-7804

RT - 1 1 1

① 認証コードに記載の2文字のアルファベットを確認の上、4桁の数字を入力してください  
アルファベットに関して

認証コードを再送 00:41秒後

② 認証コードが届かない場合

認証する

图 5: test3



## 5. 接收 & 处理 IPN(Instant Payment Notification)

发送方: G123 平台

接收方: CP

### Request Header

Request header	Description
Content-Type	application/json

### Request Body

Request Body	Description	Mandatory
orderNo	订单号	YES

平台在支付成功时发送 IPN 到 `notify_url`, 游戏服务器接收到 IPN 消息后, 应进行如下操作:

1. 立即返回 Response, 状态码为 200 OK 或 204 No Content 。若未返回 200 或 204, 平台视为发送失败, 并触发重试机制 (共 5 次)。平台会定期检查没有被正确接收的消息, 并进行重试。
2. 游戏服务器通过调用查询订单详细信息来确认订单状态: 状态为 **shipped** 时, 进行发货, 并在发货后通知 G123 平台支付状态。其他状态时, 不可发货, 也无需上报给平台。

## 6. 订单详细信息查询 API

游戏服务器在收到平台通知之后, 方可调用此接口, 禁止作为轮询使用。

### Endpoint

GET {API Host}/api/cp/orders/:orderNo

### Request Header

Request header	Description
Content-Type	application/json
Authorization	Bearer \${accessToken}

### Request Parameters

Parameters	Description
orderNo	订单号

### Response

Response	Description
orderNo	订单号
appCode	游戏 Code
status	订单状态: (shipped: 待发货/delivered: 已发货)
userId	用户 ID
paymentCode	支付方式
retryCount	重复支付次数 (失败下用户可以选择重试)
custom	创建订单时, 游戏提供参数
error	错误信息
createdAt	订单创建时间
updatedAt	订单更新时间
items	创建订单时, 游戏提供参数, Array 类型

### 用例

#### Request

```
curl -X GET \
  {API Host}/api/cp/orders/0241901200002606 \
  -H 'Authorization: Bearer K4RVtZjt1tXPSAWnADhZhZsMJ3l1fiI8' \
```

```
-H 'Content-Type: application/json' \  
-H 'cache-control: no-cache'
```

## Response

```
{  
  "orderNo": "0241901200002606",  
  "appCode": "testapp",  
  "status": "created",  
  "userId": null,  
  "paymentCode": null,  
  "retryCount": 0,  
  "custom": "{\"user\":{\"id\":\"user234\"},\"desc\":\"聖闘士星矢\"},\"serverId\":\"osaka\"",  
  "error": null,  
  "createdAt": "2019-01-20T08:26:43.000Z",  
  "updatedAt": "2019-01-20T08:26:43.000Z",  
  "items": [  
    {  
      "code": "special_bag",  
      "name": "Some Item",  
      "desc": "Some desc",  
      "amt": 100,  
      "qty": 1,  
      "taxamt": 0,  
      "url": "",  
      "createdAt": "2019-01-20T08:26:43.000Z",  
      "updatedAt": "2019-01-20T08:26:43.000Z"  
    }  
  ]  
}
```

## 7. 通知平台订单的发货成功状态

请务必确认购买的物品交付给玩家之后，或确保玩家一定能收到物品之后再上报 G123 平台发货成功

确实存在错误，导致发货失败时（如写入数据库时报错），无需上报给平台。特殊情况的处理参见常见问题

- 调用方：CP
- 提供方：G123

## Endpoint

POST {API Host}/api/cp/orders/:orderNo/deliver

## Request Header

Request header	Description
Authorization	Bearer \${{accessToken}}

## Request Parameters

Parameters	Description
orderNo	订单号

## 用例

## Request

```
curl -X POST \  
  {API Host}/api/cp/orders/订单号/deliver \  
  -H 'Authorization: Bearer K4RVtZjt1tXPSAWnADhZhZsMJ3l1fiI8' \  
  -H 'Content-Type: application/json' \  
  -H 'cache-control: no-cache'
```

## 8. 调用 G123 支付平台 API 时的错误 Response

status	Code	message	description
400	Bad access_token	Re-quest	错误
400	Bad expired_token	Re-quest	过期
400	Bad invalid_params	Re-quest	非法请求, 请检查参数; 1. JSON 格式是否正确? json validator 验证 2. notify__url 参数是否已登记到 Publisher[接收支付 IPN 通知 URL]?

## 9. 常见问题

### 1. 限购商品被多次购买

- 问题描述:

因国外在线支付特性, 用户下单到最终支付成功可能间隔较长时间, 因此有可能出现“限量商品被多次购买”的问题。游戏收到平台发货通知时, 无法向用户发放多次购买的物品。

- 解决方案:

限购商品在用户购买时让用户**明确知道是限购商品**。出现多次购买的情况, 发放等额钻石补偿, 并向平台上报发货成功。

### 2. 物品发给玩家后, 再次收到发货通知

- 问题描述:

由于网络等原因, 向平台上报发货状态时, 可能平台没有收到发货状态。这种情况下, 平台会不断重试向游戏服务器发送发货通知。

- 解决方案:

每次收到发货通知后，检查是否已经发货成功，若已经发货成功，不发货，直接调用通知平台订单的发货成功状态接口，再次向平台上报发货成功。

### 3. 用户不在线，无法收货（仅限部分游戏）

- 问题描述

部分游戏由于架构的原因，用户不在线时，无法收货。

- 解决方案

参考快递的做法，并非只有直接交给用户才算投递成功，将物品放入用户的邮箱中也是投递成功的一种方式。

因此，收到发货通知后，将订单支付成功的状态存入数据库，并确保用户下次登录能够收到物品时，即可向平台上报发货成功。

此外，目前已知有些游戏会将支付成功结果暂存在 Redis 等临时存储中，并设置超时时间，比如一个小时，造成用户在一小时内不登录游戏，就无法收货的情况。这种情况下，必须更改架构，确保用户下次登录一定能够收到物品。