import Callout from 'nextra-theme-docs/callout';

G123 CP SDK

1. 介绍

G123 平台为游戏开发商提供了一套简单易用的接口，以便在游戏中实现登录、支付等功能。本文档将介绍如何在游戏中接入 G123 平台。

2 加载 SDK

在游戏的登录页面中，加载 G123 平台提供的 SDK 文件，注意区分环境，测试环境和正式环境的 SDK 文件不同。

环境	API Host
测试	https://platform-sc.stg.g123.jp/cp-sdk/g123-cp-sdk.umd.js
正式	https://platform-sc.g123.jp/cp-sdk/g123-cp-sdk.umd.js

调用 window.CpSdk 时，需要确保 SDK 已经加载完成，因此 SDK 的加载应该当以同步的方式嵌入到页面中。

3. 类型定义 (TypeScript)

点击查看: SDK 的 TypeScript 定义

```
/**
 * 支付窗口调起参数
 *
 * 该参数由游戏服务器通过调用平台支付接口生成，然后通过游戏前端传递给 SDK
 */
```

```ts
export declare interface CreateOrderResponse {
    orderNo: string;
    token: string;
};

/**
 * 平台参数
 *
 * 用于初始化游戏，及获取 code 等参数
 */
export declare interface PlatformParams {
    /**
     * Access Code，用于获取用户信息
     */
    code: string;
    /**
     * 平台其它参数，详细参考文档
     */
    params: Record<string, string>;
    /**
     * 环境变量：'production' | 'staging'
     */
    env: string;
};

export declare interface CpSdk {
    /**
     * 获取 code 等平台参数
     */
    getPlatformParams: () => Promise<PlatformParams | undefined>;
    /**
     * 激活支付窗口
     */
```

```
    sendPayment: (params: CreateOrderResponse) => Promise<void>;
    /**
     * 保存数据
     */
    saveData: (s: string) => Promise<void>;
    /**
     * 恢复数据
     */
    restoreData: () => Promise<string>;
};

declare global {
    interface Window {
        CpSdk: CpSdk;
    }
}
```

4. API

4.1 获取平台参数

参见 账号接口 文档

此前平台会将 code 等参数通过 URL 的 query string 传递给游戏，现在改为通过 SDK 获取，游戏需要在初始化时调用 `getPlatformParams` 方法获取平台参数。

获取 code

```
const platformParams = await window.CpSdk.getPlatformParams();
// 获取 code，用于服务器端获取用户平台 ID，lang，currency 等信息
const code = platformParams.code;

// 获取其它参数
const params = platformParams.params;
```

关于其它参数，目前平台会传递以下参数：

参数名	说明	默认值
platform	主窗口 URL Query 透传，旧版 API 广告追踪参数	'ctw'
bid	主窗口 URL Query 透传，旧版 API 广告追踪参数	undefined
ver	主窗口 URL Query 透传，旧版 API 广告追踪参数	undefined
utm_medium	主窗口 URL Query 透传，用于部分游戏的活动	undefined
lang	平台语言，注意：此参数仅用于加载背景图等 critical 的资源。用户使用的语言请通过账号接口获取	'ja'
ctw_share_id	主窗口 URL Query 透传，用户通过分享链接打开时，获取分享链接的 ID	undefined
ctw_share_extra	主窗口 URL Query 透传，用户通过分享链接打开时，获取分享链接的数据	undefined

4.2 激活支付窗口

参见 支付 文档

游戏在服务器端调用平台支付 API，平台返回激活支付窗口的参数。游戏前端需要将这些参数传递给 SDK，以便 SDK 激活支付窗口。

```
const createOrderResponse = // 服务器端调用平台支付 API，返回激活支付窗口的参数

window.CpSdk.sendPayment(createOrderResponse);
```

4.3 保存和恢复数据

处理 iOS 环境下的数据存储问题

由于在 iOS 环境中，游戏 iframe 和平台 origin 不匹配，导致 localStorage 中保存的数据会在浏览器重启后丢失。为解决这个问题，请使用 SDK 的 saveData 和 restoreData 方法保存必要的数据。

注意

存储空间有限

- 平台所有游戏共享同一个 localStorage 空间，请仅保存与当前设备相关的数据，其他数据应存储在服务器上，以实现不同设备间的配置同步。
- 典型应用场景：保存用户的游戏画质设置，使用户能够在手机和 PC 端设置不同画质。

仅支持字符串格式

- 转换所有数据为字符串格式，例如使用 JSON.stringify，并合并多个数据项为单一字符串。

```
// 保存数据
const data = JSON.stringify({
    // 保存的数据
});
await window.CpSdk.saveData(data);

// 恢复数据
const restoredDataStr = await window.CpSdk.restoreData();
if (restoredDataStr) {
    // 恢复的数据
    const restoredData = JSON.parse(restoredDataStr);
}
```