

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

数据接口

G123 平台有完善的数据体系，通过获取游戏内的打点数据反应运营活动的效果，需要 CP 配合做数据上报数据上报分为两部分：1. 浏览器主窗口数据上报 2. 服务器间事件上报 - 请务必在上线前将下列数据上报给平台!!! 若没完成所有事件上报，将无法再平台上线后进行推广。

CP 校验方法: 1. Publisher 平台-> 调试工具

1. 浏览器主窗口数据上报

游戏前端需要将下面指定的日志数据，通过向浏览器主窗口 postMessage, 上报到平台。

1.1 参数

Parameter	Format	Required	Value	Example
version	string	Yes	v2	请指定版本为”v2”
action	string	Yes	g_init (其他类型参照事件类型)	用例
data	json	Yes	参照事件内容	用例

1.2 事件类型 action

下列类型的日志、请与平台数据开发人员确认埋点位置后、接入：

值	名称	调用条件
g_init	游戏初始化开始	游戏初始化代码开始执行时上报
g_register	注册游戏事件	游戏生涯首次接收到平台 ID 时上报，用户生涯仅一次。

值	名称	调用条件
g_creatorole	广告用创角事件	新玩家进入游戏后第一次点击事件时上报，如：1. 第一次游戏有开屏动画的情况下，玩家点击屏幕（下一步）或点击“跳过”（skip）时打点 2. 第一次游戏需要用户手动创角的情况下，玩家点击“创建角色”时打点 3. 新手教程中玩家根据引导进行第一次点击时打点如同时存在上述情况，请以玩家最先遇到的情况为准注：用户生涯仅一次（老用户换服无需上报）
g_login	登录游戏事件	老玩家登录游戏时上报
g_payrequest	发起支付事件	前端所有用户点击了会触发“调起支付”的地方
g_payend	支付事件结束	当玩家经由付款并发货完成时触发
g_milestone	完成里程碑事件	玩家在进行如下行为时上报：1. 玩家完成主线任务 2. 玩家挑战关卡（成功/失败）具体案例
g_click	点击对象事件	玩家在游戏内点击某埋点对象时 e.g. 玩家点击游戏内功能（如强化、突破、购买礼包）

触发逻辑线 - 【游戏启动】按时间线触发 g_init→g_register - 【新玩家】按时间线触发 g_register→g_creatorole→g_logout（新玩家注册时不要触发 g_login）- 【老玩家】按时间线触发 g_login→g_logout（老玩家登录以及换服时不要触发 g_register 和 g_creatorole）

事件细节说明 - 【新/老玩家】其他事件 g_milestone 平台的运营会根据自己的需求、提要求自定义埋点、请届时配合对应。- 【新/老玩家】其他事件 g_click 玩家最终点击礼包购买按钮之前（包含点击礼包购买按钮），所产生的全部图标点击事件。- 例 1：点击主屏幕某礼包 / 活动按钮（g_click id: 活动 id desc: 留空）-> 引导至礼包浏览页面 -> 点击某礼包的购买按钮（g_click id: 活动 id desc: 礼包 id）- 例 2：点击主屏幕商店按钮（g_click id: 商店 id desc: 礼包 id）-> 引导至某个礼包分页 -> 点击某礼包的购买按钮（g_click id: 商店 id desc: 礼包 id）- 【新/老玩家】其他事件 g_click 玩家点击强化、突破等消耗道具的功能时，道具不足导致失败时上报打点。- 例 1：点击强化按钮 -> 提示强化失败，原因缺失道具 -> 上报点击事件（g_click id: 功能 id desc: 缺失道具: 数量）

- 请注意: g_register 与 g_createrole 事件玩家生涯只能触发一次, 换服后不要重复触发。

1.3 事件内容 data

::: data 字段:::

columns	format	description	required	注
game_server_id	string	游戏区服 id	Yes	游戏服 ID, 例: 日本 12 服, 则该值为 12, 如无区服则默认为 1。始终取玩家当前的游戏服, 例如合服后取新服 ID
game_user_id	string	CP 数据库玩家账号	Yes	游戏数据库分配的账号唯一标识
item	object	item 相关字段	Yes	道具相关信息 (仅支付相关事件)
role	object	用户相关字段	Yes	玩家属性信息
custom	object	自定义字段	Yes	针对埋点对象所需传值的自定义项

::: item 字段:::

column	format	description	required	注
id	string	商品 id	Yes	埋点对象 ID, 只允许字母 (letter)、数字 (number)、- (dash) 或 _ (under-line). g_tutorial 时必须传
type	string	商品类型	Yes	menu/tab/button/gift/gacha/carc等, 根据具体事件区分, 默认为空
name	string	商品名	Yes	默认为空
count	string	商品总量	Yes	默认为空

column	format	description	required	注
price	string	商品价格	Yes	默认 为空
balance	string	商品剩余量	Yes	默认 为空
desc	string	商品描述	Yes	g_tutorial 事件 时不 能为 空

column	format	description	required	注
content	string	商品内容	Yes	商品包含的 {“{子对象的 ID}”} / {“{子对象的数量}”}, 多个子对象时前述组合以逗号 (,) 分隔, 如 IDa/2, IDb/1。 默认数量请传 1。
referer	string	商品的上一跳转名称	Yes	默认 为空
payment__type	string	付费方式, 各种虚拟货币	Yes	默认 为空

column	format	description	required	注
custom	object	item 中的其他内容	Yes	默认为空

::: role 字段:::

column	format	description	required	注
id	string	角色 id	Yes	角色在当前区服的用户角色 id
name	string	角色姓名	Yes	角色在当前区服的用户角色姓名
level	string	角色等级	Yes	
exp	string	角色经验值	Yes	
vip_level	string	角色 VIP 等级	Yes	
vip_exp	string	角色 VIP 经验	Yes	
virtual_balance	string	游戏内通货	Yes	例: “virtual_balance” : { “diamond” : 1234, “gold”: 567000 }, 根据 游戏内货 币名称填写

column	format	description	required	注
custom	object	role 中的其他内容	Yes	必传 reg_server_id, 描述如下

请注意: role.custom 中必带的一个栏位是 `reg_server_id` | column | format | value
| required| | ——| —— | ——| ——| | reg_server_id | string | 合服或换服玩家的
初始注册区服号, 无此行为的玩家默认为当前注册区服号 | Yes|

1.4 用例

用例 1: `g__init` 游戏初始化成功事件 点击查看: `g__init`

```
window.top.postMessage(JSON.stringify({
  "version": "v2",
  "action": "g_init",
  "data": {
    "custom":{
      "message": "initGame"
    }
  }
}), '*');
```

用例 2: `g__register` 用户登录事件 点击查看: `g__register`

```
window.top.postMessage(JSON.stringify({
  "version": "v2",
  "action": "g_register",
  "data": {
    "game_server_id":"1", // current server id
    "game_user_id":"1234567",
    "role": {
```



```

        "id": "3444382",
        "name": "G123 太郎",
        "level": "1",
        "exp": "1",
        "vip_level": "0",
        "vip_exp": "0",
        "virtual_balance": {
            "diamond": "1234",
            "gold": "567000"
        },
        "custom": {
            "reg_server_id": "1" // server id where g_register occurred
        }
    },
    "custom": {
        //保留字段
    }
}
}), '*');

```

用例 3: g__creatorole 用户创角事件 [点击查看: g__creatorole](#)

```

window.top.postMessage(JSON.stringify({
    "version": "v2",
    "action": "g__creatorole",
    "data": {
        "game_server_id": "1", // current server id
        "game_user_id": "1234567", //Unique user id for your game
        "role": {
            "id": "3444382", //Unique role id for the server
            "name": "G123 太郎",
            "level": "1",
            "exp": "1",
            "vip_level": "0",

```

```

        "vip_exp": "0",
        "virtual_balance": {
            "diamond": "1234",
            "gold": "567000"
        },
        "custom": {
            "reg_server_id": "1" // server id where g_register occurred
        }
    },
    "custom": {
        //保留字段
    }
}
}), '*');

```

用例 4: `g_login` 用户登录事件 点击查看: `g_login`

```

window.top.postMessage(JSON.stringify({
    "version": "v2",
    "action": "g_login",
    "data": {
        "game_server_id": "1", // current server id
        "game_user_id": "1234567",
        "role": {
            "id": "3444382",
            "name": "G123 太郎",
            "level": "1",
            "exp": "1",
            "vip_level": "0",
            "vip_exp": "0",
            "virtual_balance": {
                "diamond": "1234",
                "gold": "567000"
            },
        },
    },

```

```

        "custom": {
            "reg_server_id": "1" // server id where g_register occurred
        }
    },
    "custom":{
        //保留字段
    }
}
}), '*');

```

用例 5: g_payrequest 用户发起支付事件 点击查看: g_payrequest

```

window.top.postMessage(JSON.stringify({
    "version": "v2",
    "action": "g_payrequest",
    "data": {
        "game_server_id": "1", // current server id
        "game_user_id": "1234567",
        "role": {
            "id": "3444382",
            "name": "G123 太郎",
            "level": "1",
            "exp": "1",
            "vip_level": "0",
            "vip_exp": "0",
            "virtual_balance": {
                "diamond": "1234",
                "gold": "567000"
            },
        },
        "custom": {
            "reg_server_id": "1" // server id where g_register occurred
        }
    },
    "item":{

```

```
        "id": "",
        "type": "",
        "name": "",
        "count": "",
        "price": "",
        "balance": "",
        "desc": "",
        "content": "",
        "referer": "",
        "payment_type": "",
        "custom": {}
    },
    "custom": {
        //保留字段
    }
}
}), '*');
```

用例 6: g_payend 用户支付结束事件 点击查看: g_payend

```
window.top.postMessage(JSON.stringify({
    "version": "v2",
    "action": "g_payend",
    "data": {
        "game_server_id": "1", // current server id
        "game_user_id": "1234567",
        "role": {
            "id": "3444382",
            "name": "G123 太郎",
            "level": "1",
            "exp": "1",
            "vip_level": "0",
            "vip_exp": "0",
            "virtual_balance": {
```

```

        "diamond": "1234",
        "gold": "567000"
    },
    "custom": {
        "reg_server_id": "1" // server id where g_register occurred
    }
},
"item":{
    "id": "",
    "type": "",
    "name": "",
    "count": "",
    "price": "",
    "balance": "",
    "desc": "",
    "content": "",
    "referer": "",
    "payment_type": "",
    "custom": {}
},
"custom":{
    "order_no": "01232343434343434" // order no generated by g123
}
}
}), '*');

```

用例 7: g__milestone 用户完成里程碑事件 milestone 的格式必需遵循下面规范

```
{"milestone":{"name":" 英文描述: 对应章节"}}
```

例 1: 当用户完成任务时

```
{"milestone":{"name":"completeTask:1"}}
```

例 2: 当用户挑战章节关卡成功时

```
{"milestone":{"name":"MapProgress:1-1"}}
```

例 3: 当用户挑战章节关卡失败时

```
{"milestone":{"name":"MapFailed:1-1"}}
```

点击查看: g_milestone

```
window.top.postMessage(JSON.stringify({
  "version": "v2",
  "action": "g_milestone",
  "data": {
    "game_server_id": "1", // current server id
    "game_user_id": "1234567",
    "role": {
      "id": "3444382",
      "name": "G123 太郎",
      "level": "1",
      "exp": "1",
      "vip_level": "0",
      "vip_exp": "0",
      "virtual_balance": {
        "diamond": "1234",
        "gold": "567000"
      },
      "custom": {
        "reg_server_id": "1" // server id where g_register occurred
      }
    },
    "custom": {
      "milestone": {"name": "MapProgress:1-1"} // e.g. 通过章节1-1
    }
  }
}), '*');
```

用例 8: g_click 用户点击事件 点击查看: g_click

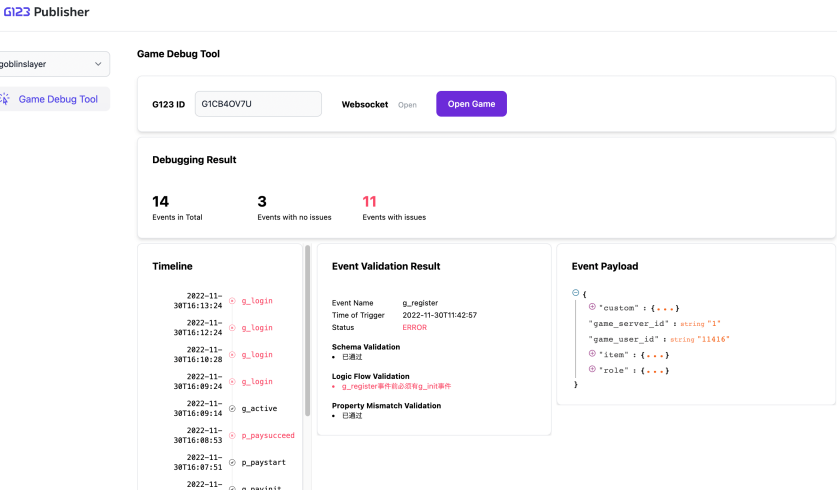
```
window.top.postMessage(JSON.stringify({
  "version": "v2",
  "action": "g_click",
  "data": {
    "game_server_id": "1", // current server id
    "game_user_id": "1234567",
    "role": {
      "id": "3444382",
      "name": "G123 太郎",
      "level": "1",
      "exp": "1",
      "vip_level": "0",
      "vip_exp": "0",
      "virtual_balance": {
        "diamond": "1234",
        "gold": "567000"
      },
      "custom": {
        "reg_server_id": "1" // server id where g_register occurred
      }
    },
    "custom": {
      "click": {
        "desc": "200005:1|20006:1", // 缺失道具 _1:数量 | 缺失道具 _2: 数量
        "id": "13", // 点击的对象 id
        "name": "qianghua" // 点击的对象名称
      }
    }
  }
}), '*');
```

1.5 测试

- 前提：账号，支付对接完成
- 登录游戏测试环境: `https://h5.stg.g123.jp/game/{appId}`, 使用平台提供的数据校验工具自查无误后，请通知平台方验收

1.5.1 数据校验工具概览

- Debugging Result: 当前游戏账号下打点校验情况总览
- Event Details: 当前所选中的打点事件详细视图显示
 - Event Payload View: 显示打点事件类型及 payload 内容
 - Event Validation Result: 是否通过打点校验，若有问题则展示错误原因，其中细分为三个分项:
 - * Schema Validation Result: 验证当前打点事件的 payload 是否符合 schema 验证条件
 - * Logic Flow Validation Result: 验证当前打点事件是否存在逻辑错误，即对于特定事件类型，不多发，不漏发，且与其他关联事件的触发时间先后顺序是否无误
 - * Property Mismatch Validation Result: 验证当前事件流的 game 相关打点事件 (g_xxx) 打点与支付相关打点事件 (p_xxx) 中, 同一 property 的值是否一致
- Timeline: 按打点事件触发时间顺序排列，顶端为最新发生的事件。黑色代表打点正确，红色代表打点错误



表打点错误

1.5.2 所支持的打点类型

- 游戏相关事件 (g_xxx): g_register, g_createrole, g_login, g_payrequest, ...etc.
- 支付相关事件 (p_xxx): p_payinit, p_paystart, p_paysucceed, ...etc.

- 工具中出现的平台触发的事件 (p_xxx), 来源于游戏的前后端打点, 或来源于 CP 接入平台接口 (账号, 支付) 时上传的数据, 故 p_xxx 事件有报错时, 请根据工具中 Timeline 的事件名描述找到相关事件的源头, 进行确认和修改

1.5.3 CP 方自查方法 CP 方技术人员若定位到 Timeline 中有问题埋点, 可以在 Timeline 上点击对应的事件, 并在 Event Validation Result 中查看问题点

Schema Validation Result Issue 例: 点击 p_paysucceed 事件, 发现错误为 must have required property 'game_user_id', 经检查 payload 发现缺失该

property

The screenshot displays the 'Game Debug Tool' interface. On the left, a sidebar shows a dropdown menu with 'gobinslayer' and a 'Game Debug Tool' button. The main area is titled 'Game Debug Tool' and includes a 'G123 ID' field with the value 'G1CB40V7U', a 'WebSocket' status indicator (Closed), and an 'Open Game' button. Below this, a 'Debugging Result' section shows statistics: 14 Events in Total, 3 Events with no issues, and 11 Events with issues. A timeline of events is listed, with the 'p_paysucceed' event at 2022-11-30T16:08:53 highlighted. To the right, the 'Event Validation Result' for 'p_paysucceed' is shown, indicating a 'Schema Validation' error: 'must have required property 'game_user_id''. Below this, the 'Event Payload' is displayed as a JSON object: { 'payment_method': 'string', 'game_server_id': 'string', 'role': {...}, 'item': {...}, 'custom': {...} }. The 'game_user_id' property is missing from the payload.

Logic Flow Validation Result Issue 例: 点击 g_register 事件, 发现错误为 g_register 事件前必须有 g_init 事件, 经检查 Timeline 发现, g_init 事件缺失

goblinslayer

Game Debug Tool

G123 IDG1CB40V7U

WebsocketClosed

Open Game

Debugging Result

14

Events in Total

3

Events with no issues

11

Events with issues

30T16:09:14

g_active

2022-11-30T16:08:53

p_paysucceed

2022-11-30T16:07:51

p_paystart

2022-11-30T16:07:50

g_payinit

2022-11-30T16:06:15

g_login

2022-11-30T16:05:44

g_login

2022-11-30T12:07:17

g_login

2022-11-30T11:45:07

g_login

2022-11-30T11:43:42

g_login

2022-11-30T11:42:57

g_register

Event Validation Result

Event Nameg_register

Time of Trigger2022-11-30T11:42:57

StatusERROR

Schema Validation

已通过

Logic Flow Validation

g_register事件前必须有g_init事件

Property Mismatch Validation

已通过

Event Payload

{

⌕

"custom" : { ... }

"game_server_id" : string "1"

"game_user_id" : string "11416"

"item" : { ... }

"role" : { ... }

}

Property Mismatch Validation Result Issue 例: 点击 p_paysucceed 事件, 发现错误为 game_server_id 与 g_levelup 事件中的值不一致, 由上面 1 的举例可知 g_levelup(来源服务器端level_up打点) 缺失 game_server_id property, 因此必然与 p_paysucceed 事件中的 game_server_id 值不一致

goblinslayer

Game Debug Tool

G123 IDG1CB40V7U

WebsocketClosed

Open Game

Debugging Result

14

Events in Total

3

Events with no issues

11

Events with issues

2022-11-30T16:12:24

g_login

2022-11-30T16:12:24

g_login

2022-11-30T16:10:28

g_login

2022-11-30T16:09:24

g_login

2022-11-30T16:09:14

g_active

2022-11-30T16:08:53

p_paysucceed

2022-11-30T16:07:51

p_paystart

2022-11-30T16:07:50

g_payinit

2022-11-30T16:06:15

g_login

Event Validation Result

Event Namep_paysucceed

Time of Trigger2022-11-30T16:08:53

StatusERROR

Schema Validation

must have required property 'game_user_id'

Logic Flow Validation

已通过

Property Mismatch Validation

game_server_id与g_levelup事件中的值不一致

Event Payload

{

⌕

"payment_method" : string "paypay"

"game_server_id" : string "1"

"role" : { ... }

"item" : { ... }

"custom" : { ... }

}

2. 服务器间事件上报

服务器端事件目前主要用于含有奖励的活动（如：升到指定级别发送 PayPay 红包），以及成果报酬型广告（如：通过第三方积分网站进入游戏，升到指定级别向用户发放第三方积分）。因此需要游戏服务器和平台服务器间通讯以确保数据可靠性，防止作弊。

2.1 升级 (level_up)

当用户达到升级时，CP 游戏服务器向平台服务器发送事件，平台返回响应消息

环境参数

系统	环境	API Host
登录	测试	https://h5.stg.g123.jp
登录	正式	https://h5.g123.jp

此外，该流程使用登录流程的 appId 和 appSecret，参见账号接口

Endpoint POST {API Host}/api/v1/cp/game_event/level_up

Request Header 使用 Basic 认证方式发送 appId 和 appSecret

Request header	Description
Content-Type	application/json
Authorization	Basic base64('appId :appSecret')

Request Body

Request			
Body	Format	Value	Required
game_server_id	string	用户所在游戏服 id	Yes
ctwid	string	g123 平台生成 id	Yes
timestamp	long	事件发生时间 (UNIX 时间单位:ms)	Yes
level	integer	用户级别	Yes
data	object	数据系统用	Yes

::: data 字段:::

columns	format	value	required	注
game_server_id	string	游戏区服 id	Yes	游戏服 ID, 例: 日本 12 服, 则该值为 12, 如无区服则默认为 1。始终取玩家当前的游戏服, 例如合服后取新服 ID
game_user_id	string	CP 数据库玩家账号	Yes	游戏数据库分配的账号唯一标识
role	object	用户相关字段	Yes	玩家属性信息
custom	object	自定义字段	Yes	针对埋点对象所需传值的自定义项, 目前必传项为 physical_server (如该 uid 触发事件所在物理服务器的域名信息)。

Request 示例 [点击查看: Request Example](#)

```
curl -i -X POST \
  -H "Content-Type:application/json" \
  -H "Authorization:Basic $(echo -n $APP_ID:$APP_SECRET | openssl base64 -A)" \
  -d \
  '{
    "game_server_id": "1",
    "ctwid": "GIL6T4C0",
    "timestamp": 1605076260715,
    "level": 10,
    "data": {
      "game_server_id": "1",
      "game_user_id": "1234567",
      "role": {
        "id": "3444382",
        "name": "G123 太郎",
        "exp": "1",
        "vip_level": "0",
        "vip_exp": "0",
        "virtual_balance": {
          "diamond": "1234",
          "gold": "567000"
        },
        "custom": {
          "reg_server_id": "1" // server id where g_register occurred
        }
      },
      "custom": {
        "device_id": "",
        "physical_server": "$SERVER_HOST|$SERVER_IP"
      }
    }
  }'
```

'https://h5.stg.g123.jp/api/v1/cp/game_event/level_up'

Response Body

- 状态码: 200
- 消息内容: JSON 格式的返回数据