# 产品设计方案

项目名称	项目编号	编写日期	编写人员
GrowAnalytics		2019-09-25	李磊

### 一、引言

### **GA系统简介**

GA全称: *growanalytics*,是针对公司内部产品特色推出的深度用户行为分析产品,支持客户端、服务器、业务数据、第三方数据的全端采集和建模,驱动营销渠道效果评估、用户精细化运营改进、产品功能机用户体验优化、实时数据监控辅助管理决策,让用户行为发挥深远的价值

### ●业务痛点

数据实时性: GA系统坚持'实时化'的设计理念,运营指标实时监控,核心指标增加预警机制,提升决策效率。

用户运营不精确: "千人一面"的全量用户营销,投入产出难以把控,不精准的粗犷方式难以真正提升存量用户的长期活跃度。GA帮助项目锁定具有相似特征的用户群体,定向推送'投其所好'的营销内容,提升不同用户群体的粘性。对长期未登录的用户群体,定向推送消息或活动召回。同时,GA建立用户标签体系和用户画像,将用户行为结合运营数据交叉分析,洞察核心用户特点,勾勒精准用户画像。

产品迭代数据量化问题: gdss41缺少实时的用户行为分析能力,使得产品功能改版的效果无法量化衡量,bug问题察觉滞后,造成用户流失。GA系统提供自助化漏斗分析模型,帮助项目组提升产品上的转化,对项目核心转化流程(如:启动、支付等)进行监控,洞察流失问题。同时,GA系统提自定义留存分析,帮助企业提高用户留存,用数据验证最受用户欢迎的产品功能,提升产品竞争力。

营销分析断层:市场营销成本居高不下,投放拉新的效果追踪出现断层,无法追踪各渠道实际转化率,难以准确分析 ROI。GA系统提供渠道追踪功能,追踪新用户的渠道来源,用数据直观评估渠道拉新能力。同时,能分析注册转化和付费转化,端到端衡量渠道拉新质量,不断调整渠道影响策略。

### ●技术痛点

日常配合效率低:项目组运营人员进行专题研究所需的数据,要向BI提需求,进入需求说明、需求评估、迭代排期、需求纠偏的循环。GA系统可帮助运营人员快速上手使用产品,无需请求数据工程师

的支持,即可通过GA系统的日日常数据看版查看日常重要数据指标;同时,通过GA系统的的自助分析模型可实现用户自定义指标查看、任意维度交叉分析、数据下钻和上卷、定位异常数据指标。

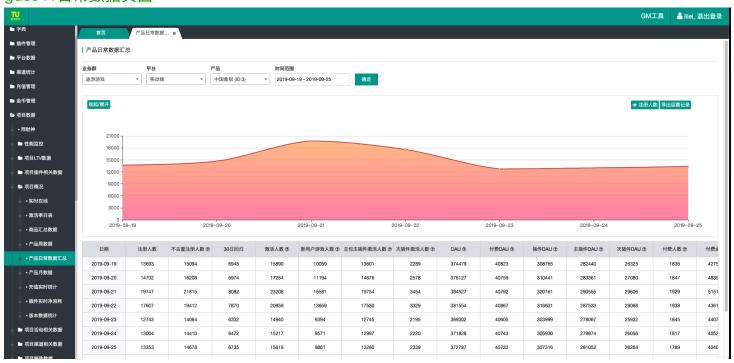
### 二、系统设计

### **心设计思想**

GrowAnalytices作为取代gdss41的新一代数据后台,除了实现原有gdss41的相关报表和数据展示功能外,更需要以灵活便捷的方式为用户提供数据查询和自动化分析能力,使用户可以根据自己的需求自助的获取相关数据完成整个数据分析留存。在前端UI美化升级的同时,赋予系统更加人性的交互方式和更加强大的系统组件为用户提供多种多样的数据获取方式。

#### 

gdss41日常数据页面

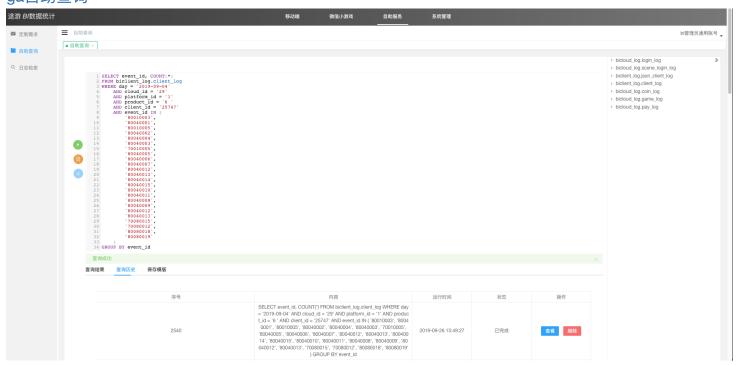


ga日常数据页面

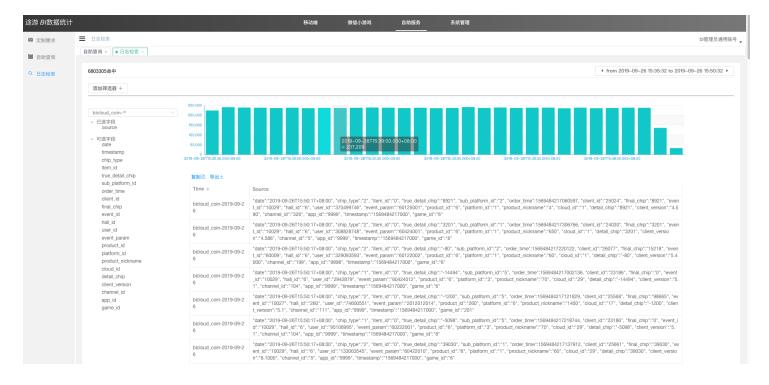


#### 🚀 ga除基本的数据展示页面外,还有多种丰富类型的页面

#### ga自助查询



ga日志检索



### **N**系统架构设计

GrowAnalytics系统采用前后端分离的开发模式,主要由前端、中间层及后端微服务构成。其中前端及中间层部分由BM负责开发,可以完整的支撑整个系统的基本流程,BI负责开发后端接口只需面对业务逻辑即可。

类型	技术栈	功能
客户端	vue、element-ui	客户端页面展示
中间层	php、swoole	服务的渲染、websocket服务、用户及权限校验
服务端	spring-cloud、spring-boot	数据核心功能服务

#### 1、前端

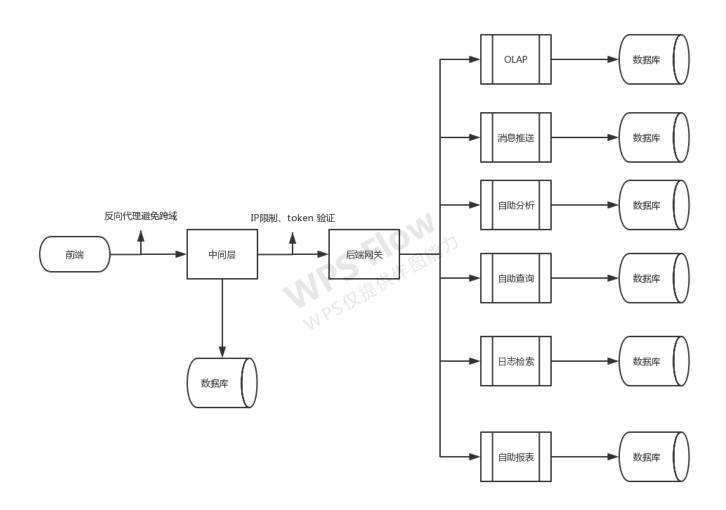
前端使用vue框架以及配套的vuex和vue-router组件,实现前端路由跳转以及前端缓存等相关功能,接口请求使用axios组件,数据格式统一使用json格式。页面渲染搭配使用element-ui完成页面基础组件的实现,可以灵活美观的构建所需要的复杂页面需求。

#### 2、中间层

中间层主要负责进行服务端渲染,建立服务端和前端交互的桥梁,屏蔽了中间的复杂性和反复校验环节。利用php+swoole实现websocket长连接,为前后端的交互提供http以外的额外功能,实现页面交互的异步刷新和后端消息主动推送功能。

#### 3、服务端

spring-cloud实现的微服务集群,由后端开发者提供整个系统核心的服务部分,完成核心的数据服务部分。



### **N**系统分析

在GrowAnalytics中用户的整个请求过程涉及多层间的交互,加上bi数据处理中可能出现加载超时等问题,天然具备不同于普通web系统复杂性。因此在整个系统的实现中,引入websokcet处理常规http服务之外的需求,在部分细节处理中也做出特殊设计以满足交互需求。

#### 1、用户接口请求校验

用户登录状态通过jwt实现,在用户登录后将用户必要信息序列化后通过des3加密生成用户token返回给前端,在后续的每次接口请求中都必须携带该token通过用户校验才可以正常访问。

#### jwt格式:

```
{
    userName: "admin", // 用户名
    permissions: [], // 用户权限
    expire: 15885466411 // 过期时间(毫秒时间戳)
}
```

#### 2、websocket的使用

用户正常登录后前端初始化socket连接,在用户登录状态超时前长连接始终保持连接状态,过程中一旦因为网络波动或其他不可抗力因素断开连接只需重连即可。长链持续中后端数据处理成功后可以通过调用中间层的接口实现对前端的主动推送功能。

### 三、编码规范

### ◎前后端通信接口规范

#### 1、接口格式

#### 1-1、基本格式

```
{
    "status": 200,
    "msg": "success",
    "data": {}
}
```

- status: 接口状态码
- msg: 接口返回消息
- data: 接口中提供的数据集合(可以是任意层的嵌套)

ps:在操作返回的接口中由于没有需要回传的数据,所以在此类接口中data设为空即可

#### 1-2、数据列表获取接口

```
{
    "status": 200,
    "msg": "success",
    "data": {
        "content": [],
        "totalElements": 0
    }
}
```

- content: 数据记录结果, 一般是结果数组
- totalElements: 当前条件下总记录条数由于分页的原因一次只会在items中返回有限条数

ps:由于前端分页组件普遍需要数据总条数,所以在data中一般将返回的数据列放入items中,total 用于返回总记录条数

#### 1-3、操作结果返回接口

```
{
    "status": 404,
    "msg": "Not Found",
    "data": {}
}

{
    "status": 403,
    "msg": "权限不足"
}
```

ps: 在此类接口中msg只用于显示报错信息,若有其他需要回传处理的数据都加入到data中即可

#### 2、接口状态码规范

状态码	类型	描述	
2**	success	接口正常返回无错误	
4**	exception	接口异常,提示信息反馈给用户	
5**	error	系统错误,对用户屏蔽报错信息	

ps: 在日常使用中,若有特殊需求需自定义状态码时,应尽量避免与已有通用httpcode码冲突

### 外部统计接口规范

对外接口调用httpMethod主要涉及get、post、put、delete四种方式,其中get与delete为简单调用方式参数构建到URL中通过明文传递。post和put数据放入body中以json格式传递,http到头信息中会通过Content-typ:application/json;charset=utf-8进行标注。

# **◇系统代码管理svn(git)**

系统开发过程中采用svn+git的方式进行代码版本管理,由于前端框架中会产生大量不需要加入版本管理的资源,相对于svn而言git只需要使用.gitignore文件即可便捷筛选管理。因此代码日常的变动管理使用公司内网的gitLab进行管理,上线时将打包好的项目通过svn上传部署到线上服务器即可。

#### へ gitFlow工作流图例



### ◎项目部署

#### 前端

使用nginx直接配置代理dist目录即可,通过indx.html入口即可进入系统。

#### 后端

将代码打包后产出的simple.phar拉取到线上,通过php simple.phar start 即可启动进程,此处使用 supervisor进程管理工具进行托管防止应用进程意外挂起。其次通过nginx反向代理将后端监听端口转发 到前端域名下,绕开浏览器同源策略问题实现接口同源化避免跨域问题。

### 四、功能模块设计

# ◇用户登录及权限管理

#### 功能用途

实现系统基本的登录和权限控制功能,在接入实际业务模块前可以流畅的概览整个系统基本框架结构。

#### 数据库设计

```
/*
Navicat Premium Data Transfer
Source Server
                    : 开发测试
Source Server Type : MySQL
Source Server Version: 50723
 Source Host
             : 192.168.10.143
 Source Database : grow_analytics
Target Server Type : MySQL
Target Server Version: 50723
 File Encoding : utf-8
Date: 10/28/2019 16:24:23 PM
SET NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;
-- Table structure for `system_game`
DROP TABLE IF EXISTS `system_game`;
CREATE TABLE `system_game` (
  `id` int(10) unsigned NOT NULL AUTO INCREMENT COMMENT '主键id',
  `name` varchar(50) CHARACTER SET utf8 DEFAULT '默认游戏' COMMENT '游戏名称',
  `icon` text CHARACTER SET utf8 COMMENT '游戏图标',
  `platform_id` int(10) unsigned DEFAULT '1' COMMENT '应用类型 1移动端 3微信',
  `app_id` int(10) unsigned DEFAULT '1' COMMENT '应用编码',
  `time zone` varchar(50) DEFAULT '+8' COMMENT '游戏数据统计的时区',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '状态 0关闭 1开启',
  `updater` varchar(100) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '修改人',
  `update_time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '修改时间',
  `creater` varchar(255) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '创建人',
  `create_time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '创建时间',
 PRIMARY KEY ('id'),
 KEY `index` (`name`,`platform_id`,`app_id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COMMENT='游戏列表';
— Table structure for `system_invitation`
DROP TABLE IF EXISTS `system_invitation`;
CREATE TABLE `system invitation` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT COMMENT '主键id',
  `name` varchar(50) CHARACTER SET utf8 DEFAULT 'name' COMMENT '姓名',
  `user name` varchar(100) CHARACTER SET utf8 DEFAULT 'ga@tuyoogame.com' COMMENT '用户账'
  `level` int(10) unsigned DEFAULT '100' COMMENT '用户角色等级',
  `permission` json DEFAULT NULL COMMENT '分配的权限树',
  `url` text CHARACTER SET utf8 COMMENT '邀请链接',
  `status` tinyint(1) unsigned DEFAULT '0' COMMENT '状态 0未接受 1已发送 2已接受',
```

```
`creater` varchar(50) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '创建人',
  `create_time` datetime DEFAULT '1000-01-01 00:00:00' ON UPDATE CURRENT_TIMESTAMP COMMI
 PRIMARY KEY ('id')
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4;
-- Table structure for `system_map`
DROP TABLE IF EXISTS `system_map`;
CREATE TABLE `system_map` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(100) CHARACTER SET utf8 DEFAULT '测试字典' COMMENT '权限名称',
  `ext` varchar(100) DEFAULT 'permission' COMMENT '字典编码(唯一标识)',
  `type` varchar(50) DEFAULT 'admin' COMMENT '字典类型',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '状态 0关闭 1开启',
  `updater` varchar(50) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '修改人',
  `update_time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '修改时间',
  `creater` varchar(255) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '创建人',
  `create_time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '创建时间',
 PRIMARY KEY ('id'),
 KEY `index` (`name`, `ext`, `type`)
) ENGINE=InnoDB AUTO_INCREMENT=53 DEFAULT CHARSET=utf8mb4 COMMENT='系统关键字映射字典';
-- Table structure for `system_permission`
DROP TABLE IF EXISTS `system permission`;
CREATE TABLE `system permission` (
  `id` int(10) unsigned NOT NULL AUTO INCREMENT,
  `name` varchar(100) CHARACTER SET utf8 DEFAULT '默认权限' COMMENT '权限名称',
  `ext` varchar(100) DEFAULT 'permission' COMMENT '权限编码(唯一标识)',
  `type` varchar(50) DEFAULT 'admin' COMMENT '权限类型',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '状态 0关闭 1开启',
  `updater` varchar(50) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '修改人',
  `update time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '修改时间',
  `creater` varchar(255) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '创建人',
  `create_time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '创建时间',
 PRIMARY KEY ('id'),
 KEY `index` (`name`, `ext`, `type`)
) ENGINE=InnoDB AUTO INCREMENT=8 DEFAULT CHARSET=utf8mb4 COMMENT='权限列表';
-- Table structure for `system_studio`
DROP TABLE IF EXISTS `system studio`;
CREATE TABLE `system_studio` (
  `id` int(10) unsigned NOT NULL AUTO INCREMENT COMMENT '主键id',
  `name` varchar(100) CHARACTER SET utf8 DEFAULT 'studio' COMMENT '工作室名称',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '状态 0关闭 1开启',
  `updater` varchar(100) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '修改人',
  `update_time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '修改时间',
```

```
`creater` varchar(100) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '创建人',
  `create_time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '创建时间',
 PRIMARY KEY ('id')
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COMMENT='工作室表';
— Table structure for `system_studio_game`
DROP TABLE IF EXISTS `system_studio_game`;
CREATE TABLE `system_studio_game` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT COMMENT '主键id',
  `studio_id` int(10) unsigned DEFAULT '1' COMMENT '工作室id',
  `game_id` int(10) unsigned DEFAULT '1' COMMENT '游戏id',
 PRIMARY KEY ('id'),
 KEY `index` (`game_id`,`studio_id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COMMENT='工作室与管理员关系映射表一
-- Table structure for `system_user`
DROP TABLE IF EXISTS `system_user`;
CREATE TABLE `system_user` (
  `user_name` varchar(100) NOT NULL DEFAULT 'admin' COMMENT '用户登录账号',
  `password` char(32) DEFAULT '123456' COMMENT '密码',
  `name` varchar(100) CHARACTER SET utf8 DEFAULT '管理员' COMMENT '用户名称',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '状态 0关闭 1开启',
  `level` tinyint(10) unsigned DEFAULT '100' COMMENT '用户角色等级 0BI管理员 10项目运营 20测
  `is admin` tinyint(1) unsigned DEFAULT '0' COMMENT '是否是bi管理员',
  `updater` varchar(100) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '修改人',
  `update time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '修改时间',
  `creater` varchar(100) CHARACTER SET utf8 DEFAULT 'admin' COMMENT '创建人',
  `create time` datetime DEFAULT '1000-01-01 00:00:00' COMMENT '创建时间',
 PRIMARY KEY (`user name`),
 KEY `unio` (`user_name`,`password`,`level`,`status`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='用户表':
— Table structure for `system_user_permission`
DROP TABLE IF EXISTS `system_user_permission`;
CREATE TABLE `system_user_permission` (
  `id` int(10) unsigned NOT NULL AUTO INCREMENT COMMENT '主键id',
  `user_name` varchar(100) CHARACTER SET utf8 NOT NULL DEFAULT 'admin@tuyoogame.com' CON
  `studio_id` int(10) unsigned NOT NULL DEFAULT '1' COMMENT '工作室id',
  `distribute name` varchar(100) CHARACTER SET utf8 DEFAULT 'admin@tuyoogame.com' COMMEN
  `permission` json DEFAULT NULL COMMENT '权限列表',
  `game` json DEFAULT NULL COMMENT '游戏列表',
  `is auto` tinyint(1) unsigned DEFAULT '0' COMMENT '是否自动填充工作室新权限',
  `is_distribute` tinyint(1) unsigned DEFAULT '0' COMMENT '是否可进行分配',
  `is_access_game` tinyint(1) DEFAULT '0' COMMENT '是否可以接入游戏',
  `is_admin` tinyint(1) unsigned DEFAULT '0' COMMENT '是否是工作室管理员',
```

```
`status` tinyint(1) unsigned DEFAULT '0' COMMENT '权限激活状态 0未激活 1已激活', PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COMMENT='用户持有权限列表--包含授权
SET FOREIGN_KEY_CHECKS = 1;
```

#### 接口文档

# system

系统基础接口,用户和权限等

# 登录注册

### 权限分发

### 基本信息

Path: /system/login/openPermission

Method: GET

接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
token	是		权限令牌

# 用户注册

### 基本信息

Path: /sign

Method: POST

接口描述:

# 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### Body

参数名称	参数类型	是否必须	示例	备注
user_name	text	是	admin	账号
password	text	是	123456	密码
name	text	是	管理员	用户名称
level	text	否	40	用户角色等级

# 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码	mock: @id
msg	string	必须		反馈消息	mock: @ctitle
data	number	必须			

# 用户登录

# 基本信息

Path: /login

Method: GET

接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
user_name	是	admin	登录名
password	是	123456	密码

# 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须			mock: @guid
msg	string	必须			mock: @ctitle
data	object	必须			
token	string	必须		用户登录凭证token	mock: @string

# 用户管理

# 删除用户的可控权限

### 基本信息

Path: /system/user/delUserPermission

Method: GET

接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
user_name	是	lilei@tuyoogame.com	要删除的用户账号

# 新增用户并分配权限

### 基本信息

Path: /system/user/createPermission

Method: POST

接口描述:

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### **Body**

参数名称	参数类型	是否必须	示例	备注
name	text	是	李磊	用户名称
user_name	text	是	lilei@tuyoogame.com	用户账号
level	text	是	40	用户角色等级
permission	text	是		分配的权限列表

# 用户列表

### 基本信息

Path: /system/user

Method: GET

接口描述:

# 请求参数

### Query

参数名称	是否必须	示例	备注
name[like]	否	管理员	用户名称(模糊匹配)
_limit[page]	否	1	页码
_limit[size]	否	20	每页条数

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object	非必须			
items	object	非必须			item 类型: object
- user_name	string	非必须		账号	
- password	string	非必须		密码	
- name	string	非必须		用户名称	
status	number	非必须		状态 0关闭 1开启	
level	number	非必须		用户角色等级	
- updater	string	非必须		修改人	

名称	类型	是否必须	默认值	备注	其他信息
 update_time	string	非必须		最后修改时间	
- creater	string	非必须		创建人	
- create_time	string	非必须		创建时间	
total	number	非必须		总记录数	

# 用户添加

# 基本信息

Path: /system/user

Method: POST

接口描述:

# 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

### Body

参数名称	参数类型	是否必须	示例	备注
user_name	text	是	admin	账号
password	text	是	123456	密码
name	text	是	管理员	用户名称

名称 类型 是否必须	默认值	备注	其他信息
------------	-----	----	------

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码	mock: @id
msg	string	必须		反馈消息	mock: @ctitle
data	number	必须			

# 编辑用户权限

### 基本信息

Path: /system/user/setUserPermission

Method: PUT

接口描述:

### 请求参数

#### **Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### Body

参数名称	参数类型	是否必须	示例	备注
user_name	text	是	lilei@tuyoogame.com	用户账号
permission	text	是		权限回传列表
name	text	是		用户名称
level	text	是		用户角色等级

名称	类型	是否必须	默认值	备注	其他信息
----	----	------	-----	----	------

# 获取用户信息

# 基本信息

Path: /userInfo

Method: GET

接口描述:

# 请求参数

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须			mock: @id
msg	string	必须			mock: @title
data	object	必须			
- user_name	string	必须		用户名	mock: @name
- name	string	必须		用户名称	mock: @cname
- is_distribute	string	必须		是否可分配权限 (管理用户)	mock: @boolean
 is_access_game	string	必须		是否可接入游戏	mock: @boolean
- permission	string []	必须		用户所有的游戏和权限	item 类型: string

名称	类型	是否必须	默认值	备注	其他信息
<b>-</b>		非必须			

# 获取用户可分配权限列表

### 基本信息

Path: /system/user/getPermissionMap

Method: GET

接口描述:

### 请求参数

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object []	非必须			item 类型: object
id	number	非必须		工作室id	
- name	string	非必须		工作室名称	
- game	object []	非必须		游戏列表	item 类型: object
id	string	必须		游戏id	
- name	string	必须		游戏名称	
- permission	object []	非必须		权限列表	item 类型: object
- type	string	必须		权限类型	
- value	object []	必须		权限列	item 类型: object

名称	类型	是否必须	默认值	备注	其他信息
├— name	string	必须		权限名称	
├— ext	string	必须		权限编码	
├— type	string	必须		权限类型	

# 获取编辑用户权限回显列表

### 基本信息

Path: /system/user/getEditPermissionMap

Method: GET

接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
user_name	是	lilei@tuyoogame.com	用户账号

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object []	非必须			item 类型: object
id	number	非必须		工作室id	
- name	string	非必须		工作室名称	
- game	object []	非必须		游戏列表	item 类型: object

名称	类型	是否必须	默认值	备注	其他信息
id	string	必须		游戏id	
├— name	string	必须		游戏名称	
- permission	object []	非必须		权限列表	item 类型: object
type	string	必须		权限类型	
- value	object []	必须		权限列	item 类型: object
- name	string	必须		权限名称	
├— ext	string	必须		权限编码	
- type	string	必须		权限类型	

# 工作室管理

# 工作室修改

# 基本信息

Path: /system/studio

**Method:** PUT

接口描述:

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### **Body**

参数名称	参数类型	是否必须	示例	备注
id	text	是	1	工作室id
name	text	否	测试工作室	工作室名称
admin	text	是	李磊	工作室管理员名称
user_name	text	否	lilei@tuyoogame.com	工作室管理员账号
status	text	否	1	状态 0关闭 1开启 默认开启

# 返回数据

名称 类型	是否必须 默认值	备注    其他	信息
-------	----------	----------	----

# 工作室列表

### 基本信息

Path: /system/studio

**Method:** GET

接口描述:

# 请求参数

#### Query

参数名称	是否必须	示例	备注
name[like]	否	测试工作室	工作室名称(模糊匹配)
_limit[page]	否	1	分页页码
_limit[size]	否	20	每页条数

名称	类型	是否必须	默认值	备注	其他信息
----	----	------	-----	----	------

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须			
msg	string	必须			
data	object	必须			
├ items	object	必须		数据集合	item 类型: object
- name	string	必须		工作室名称	
status	number	必须		状态 0关闭 1开启	
- game	object	必须		游戏列表	item 类型: object
├— id	string	必须		游戏id	
- name	string	必须		游戏名称	
- admin	string []	必须		管理员列表	item 类型: string
<b>├</b>		非必须			
updater	string	必须		修改人	
 update_time	string	必须		最后修改时间	
creater	string	必须		创建人	
create_time	string	必须		创建时间	
total	number	必须		总条数	

# 工作室添加

# 基本信息

Path: /system/studio

Method: POST

接口描述:

### 请求参数

#### **Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### **Body**

参数名称	参数类型	是否必须	示例	备注
name	text	是	第一工作室	工作室名称
admin	text	是	李磊	工作室管理员名称
user_name	text	是	lilei@tuyoogame.com	工作室管理员账号
status	text	否	1	状态 0关闭 1开启 默认开启

# 返回数据

名称
----

# 工作室详情

### 基本信息

Path: /system/studio/1

Method: GET

接口描述:

路由最后的数字是工作室id

# 请求参数

### 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object	非必须			
id	number	非必须		工作室id	
- name	string	非必须		工作室名称	
- user_name	string	非必须		管理员账号	
- admin	string	非必须		管理员名称	

# 游戏管理

# 游戏接入

### 基本信息

Path: /system/game

Method: POST

接口描述:

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### Body

参数名称	参数类型	是否必须	示例	备注
------	------	------	----	----

参数名称	参数类型	是否必须	示例	备注
studio_id	text	是	1	工作室id
platform_id	text	是	1	应用类型
app_id	text	是	251450	应用id
time_zone	text	是	+8	时区
icon	text	是	aaa.jpg	图标
name	text	是	斗地主	游戏名称

### 返回数据

名称	类型	是否必须	默认值	备注	其他信息	

# 权限内容管理

# 权限修改

### 基本信息

Path: /system/permission

Method: PUT

接口描述:

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### Body

参数名称	参数类型	是否必须	示例	备注
id	text	是	1	权限id
name	text	否	添加或编辑团队成员	权限名称
ext	text	否	user_sit	权限编码
type	text	否	admin	权限分类
status	text	是	1	状态 0关闭 1开启 默认开启

# 权限列表

# 基本信息

Path: /system/permission

Method: GET

接口描述:

# 请求参数

#### Query

参数名称	是否必须	示例	备注
name[like]	否	编辑	权限名称(模糊匹配)
type	否	admin	权限类型
status	否	1	状态 0关闭 1开启
_limit[page]	否	1	页码
_limit[size]	否	20	每页条数

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须			

名称	类型	是否必须	默认值	备注	其他信息
msg	string	必须			
data	object	必须			
├— items	object []	必须			item 类型: object
├— id	number	必须		权限id	
- name	string	必须		权限名称	
├— ext	string	必须		权限编码	
	string	必须		权限类型	
- status	number	必须		状态	
- updater	string	必须		修改人	
- update_time	string	必须		最后修改时间	
creater	string	必须		创建人	
- create_time	string	必须		创建时间	
total	number	必须		总条数	

# 权限添加

# 基本信息

Path: /system/permission

Method: POST

接口描述:

# 请求参数

#### Headers

参数名称    参	数值     是	是否必须 示例	备注
-----------	----------	---------	----

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### Body

参数名称	参数类型	是否必须	示例	备注
name	text	是	编辑应用设置页面	权限名称
ext	text	是	page_sit	权限编码
type	text	是	admin	权限类型
status	text	否	1	状态 0关闭 1开启 默认开启

# 权限详情

# 基本信息

Path: /system/permission/1

Method: GET

接口描述:

# 请求参数

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须			
msg	string	必须			
data	object	必须			
id	number	必须		权限id	
- name	string	必须		权限名称	
├— ext	string	必须		权限编码	

名称	类型	是否必须	默认值	备注	其他信息
- type	string	必须		权限类型	
- status	number	必须	状态 0关闭 1开启		
- updater	string	必须		修改人	
- update_time	string	必须		最后修改时间	
- creater	string	必须		创建人	
- create_time	string	必须		创建时间	

# 获取权限映射

### 基本信息

Path: /system/permission/getMap

Method: GET

接口描述:

### 请求参数

Query

参数名称	是否必须	示例	备注
type	是	admin	权限类型

# 字典管理

# 修改字典

### 基本信息

Path: /system/map

Method: PUT

接口描述:

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### **Body**

参数名称	参数类型	是否必须	示例	备注
id	text	是	1	权限id
name	text	否	添加或编辑团队成员	权限名称
ext	text	否	user_sit	权限编码
type	text	否	admin	权限分类
status	text	是	1	状态 0关闭 1开启 默认开启

# 字典列表

### 基本信息

Path: /system/map

Method: GET

接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
name[like]	否	编辑	权限名称(模糊匹配)

参数名称	是否必须	示例	备注
type	否	admin	权限类型
status	否	1	状态 0关闭 1开启
_limit[page]	否	1	页码
_limit[size]	否	20	每页条数

# 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须			
msg	string	必须			
data	object	必须			
- items	object []	必须			item 类型: object
id	number	必须		权限id	
- name	string	必须		权限名称	
ext	string	必须		权限编码	
type	string	必须		权限类型	
- status	number	必须		状态	
- updater	string	必须		修改人	
- update_time	string	必须		最后修改时间	
- creater	string	必须		创建人	
- create_time	string	必须		创建时间	
- total	number	必须		总条数	

# 字典映射获取

# 基本信息

Path: /system/map/getMap

Method: GET

接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
type	是	role	字典类型

# 返回数据

名称	类型	是否必须	默认值	备注	其他信息	
----	----	------	-----	----	------	--

# 字典详情

# 基本信息

Path: /system/map/1

Method: GET

接口描述:

### 请求参数

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须			
msg	string	必须			
data	object	必须			
id	number	必须		权限id	

名称	类型	是否必须	默认值	备注	其他信息
- name	string	必须		权限名称	
├— ext	string	必须		权限编码	
	string	必须		权限类型	
- status	number	必须		状态 0关闭 1开启	
- updater	string	必须		修改人	
- update_time	string	必须		最后修改时间	
- creater	string	必须		创建人	
- create_time	string	必须		创建时间	

# 添加字典

### 基本信息

Path: /system/map

Method: POST

接口描述:

# 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### Body

参数名称	参数类型	是否必须	示例	备注
name	text	是	编辑应用设置页面	权限名称
ext	text	是	page_sit	权限编码

参数名称	参数类型	是否必须	示例	备注
type	text	是	admin	权限类型
status	text	否	1	状态 0关闭 1开启 默认开启

# 获取下拉框填充数据

### 基本信息

Path: /system/map/getSelect

Method: GET

接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
type	是	role	字典类型

# 获取用户可接入游戏的工作室下拉框

### 基本信息

Path: /system/map/getUserStudioSelect

Method: GET

接口描述:

### 请求参数

# 文件上传

## 图片上传

#### 基本信息

Path: /system/file/uploadImg

Method: POST

接口描述:

#### 请求参数

#### **Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	multipart/form-data	是		

#### Query

参数名称	是否必须	示例	备注
file	是		

#### **Body**

参数名称	参数类型	是否必须	示例	备注
file	file	是		

# **WebSocket**

### 消息弹窗

#### 基本信息

Path: /system/socket/send

Method: GET

#### 接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
user_name	是		用户账号
method	是	message提示消息 notify推送消息	触发行为类型
title	否		推送消息框标题
content	否		消息内容
type	否		弹框类型
time	否		弹框驻留事件

### 返回数据

名称	类型	是否必须	默认值	备注	其他信息

### **沙数据服务配置**

#### 功能用途

为自助分析模块的检索选项增添和管理相关选项,提供下拉列表相关接口。

#### 数据库设计

```
/*
Navicat Premium Data Transfer
Source Server
                     : 开发测试
Source Server Type : MySQL
Source Server Version: 50723
Source Host
                    : 192.168.10.143
Source Database : data_services_config
Target Server Type : MySQL
Target Server Version: 50723
 File Encoding : utf-8
Date: 10/28/2019 16:24:08 PM
SET NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;
-- Table structure for `28-device_property`
DROP TABLE IF EXISTS `28-device_property`;
CREATE TABLE `28-device_property` (
  `name` varchar(50) NOT NULL COMMENT '属性名',
  `show_name` varchar(100) DEFAULT NULL COMMENT '显示名',
  `data type` varchar(50) DEFAULT NULL COMMENT '数据类型',
  `format` varchar(50) DEFAULT NULL COMMENT '单位/格式',
  `dictionaries` varchar(100) DEFAULT NULL COMMENT '字典',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '显示状态',
 `remark` text COMMENT '备注',
  `preset` tinyint(1) unsigned DEFAULT '1' COMMENT '预置',
  `position` varchar(100) DEFAULT NULL COMMENT '字段位置',
 PRIMARY KEY (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
— Table structure for `28-event`
DROP TABLE IF EXISTS `28-event`:
CREATE TABLE `28-event` (
  `name` varchar(50) NOT NULL COMMENT '事件名',
  `show_name` varchar(50) DEFAULT NULL COMMENT '显示名',
  `label` varchar(50) DEFAULT NULL COMMENT '标签',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '显示状态',
  `remark` text COMMENT '备注',
 PRIMARY KEY (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-- Table structure for `28-event_property`
```

```
DROP TABLE IF EXISTS `28-event_property`;
CREATE TABLE `28-event_property` (
  `name` varchar(50) NOT NULL COMMENT '属性名',
  `show_name` varchar(100) DEFAULT NULL COMMENT '显示名',
  `data_type` varchar(50) DEFAULT NULL COMMENT '数据类型',
  `format` varchar(50) DEFAULT NULL COMMENT '单位/格式',
  `dictionaries` varchar(100) DEFAULT NULL COMMENT '字典',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '显示状态',
  `remark` text COMMENT '备注',
  `preset` tinyint(1) unsigned DEFAULT '1' COMMENT '预置',
  `position` varchar(100) DEFAULT NULL COMMENT '字段位置',
  PRIMARY KEY (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
  Table structure for `28-event_property_map`
DROP TABLE IF EXISTS `28-event_property_map`;
CREATE TABLE `28-event_property_map` (
  `event_name` varchar(100) NOT NULL COMMENT '事件名称',
  `property_name` varchar(100) NOT NULL COMMENT '属性名',
 PRIMARY KEY (`event_name`, `property_name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
— Table structure for `28-user_property`
__ _____
DROP TABLE IF EXISTS `28-user_property`;
CREATE TABLE `28-user_property` (
  `name` varchar(50) NOT NULL COMMENT '属性名',
  `show name` varchar(100) DEFAULT NULL COMMENT '显示名',
  `data type` varchar(50) DEFAULT NULL COMMENT '数据类型',
  `format` varchar(50) DEFAULT NULL COMMENT '单位/格式',
  `dictionaries` varchar(100) DEFAULT NULL COMMENT '字典',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '显示状态',
  `remark` text COMMENT '备注',
  `preset` tinyint(1) unsigned DEFAULT '1' COMMENT '预置',
  `position` varchar(100) DEFAULT NULL COMMENT '字段位置',
  PRIMARY KEY (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
  Table structure for `28-virtual_event`
DROP TABLE IF EXISTS `28-virtual_event`;
CREATE TABLE `28-virtual event` (
  `name` varchar(50) NOT NULL COMMENT '事件名',
  `show_name` varchar(50) DEFAULT NULL COMMENT '显示名',
  `label` varchar(50) DEFAULT NULL COMMENT '标签',
  `status` tinyint(1) unsigned DEFAULT '1' COMMENT '显示状态',
```

```
`remark` text COMMENT '备注',
`event_list` json DEFAULT NULL COMMENT '包含事件',
`value` json DEFAULT NULL COMMENT '虚拟事件json片段',
PRIMARY KEY (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

SET FOREIGN_KEY_CHECKS = 1;
```

#### 接口文档

# data\_services\_config

# 虚拟事件

### 添加虚拟事件

#### 基本信息

Path: /config/virtualEvent

Method: POST

接口描述:

#### 请求参数

#### **Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

参数名称	参数类型	是否必须	示例	备注
name	text	是		事件名

参数名称	参数类型	是否必须	示例	备注
show_name	text	是		显示名
label	text	是		标签
value	text	是		事件内容
remark	text	否		备注

# 虚拟事件列表

### 基本信息

Path: /config/virtualEvent

Method: GET

接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
_limit[page]	是	1	页码
_limit[size]	是	20	每页条数

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object	非必须			
- items	object []	非必须			item 类型: object
├— name	string	非必须		事件名	

名称	类型	是否必须	默认值	备注	其他信息
show_name	string	非必须		显示名称	
label	string	非必须		标签	
status	number	非必须		显示状态	
remark	string	非必须		备注	
event_list	string []	非必须		包含原事件	item 类型: string
<b></b>		非必须			
- value	null	非必须			
total	number	非必须		总条数	

# 虚拟事件编辑

### 基本信息

Path: /config/virtualEvent

**Method:** PUT

接口描述:

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

参数名称	参数类型	是否必须	示例	备注
name	text	是		事件名
show_name	text	否		显示名

参数名称	参数类型	是否必须	示例	备注
label	text	否		标签
value	text	否		事件内容
remark	text	否		备注

### 虚拟事件详情

#### 基本信息

Path: /config/virtualEvent/getInfo

Method: GET

接口描述:

#### 请求参数

#### Query

参数名称	是否必须	示例	备注
name	是		事件名

# 下拉框

## 事件下拉列表

#### 基本信息

Path: /config/select/event

Method: GET

接口描述:

#### 请求参数

#### Query

参数名称	是否必须	示例	备注
virtual	否	true	是否包含虚拟事件

#### 返回数据

名称 类型	是否必须	默认值	备注	其他信息	
-------	------	-----	----	------	--

## 属性下拉列表

#### 基本信息

Path: /config/select/property

Method: GET

接口描述:

#### 请求参数

### 属性数据类型符号获取

#### 基本信息

Path: /config/select/symbol

Method: GET

接口描述:

#### 请求参数

# 事件属性

# 事件属性列表

### 基本信息

Path: /config/eventProperty

Method: GET

接口描述:

### 请求参数

#### Query

参数名称	是否必须	示例	备注
_limit[page]	否		页码
_limit[size]	否		每页记录数
name[like]	否		属性名搜索

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object	非必须			
- items	object []	非必须			item 类型: object
name	string	非必须		属性名	
show_name	string	非必须		显示名	
is_update	number	非必须		是否可修改	
- data_type	string	非必须		数据类型	
format	string	非必须		单位/格式	
- dictionaries	string	非必须		字典	

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须		显示状态	
- preset	number	非必须		预置	
- position	string	非必须		字段位置	
total	number	非必须		总记录数	

# 事件属性编辑

### 基本信息

Path: /config/eventProperty

Method: PUT

接口描述:

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

参数名称	参数类型	是否必须	示例	备注
name	text	是		属性名
show_name	text	否		显示名
data_type	text	否		数据类型
format	text	否		单位/格式
dictionaries	text	否		字典
status	text	否		显示状态

参数名称	参数类型	是否必须	示例	备注
remark	text	是		备注

名称	类型	是否必须	默认值	备注	其他信息

# 事件属性详情

### 基本信息

Path: /config/eventProperty/getInfo

Method: GET

接口描述:

#### 请求参数

#### Query

参数名称	是否必须	示例	备注
name	否		属性名

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object	非必须			
name	string	非必须		属性名	
show_name	string	非必须		显示名	
├— is_update	number	非必须			

名称	类型	是否必须	默认值	备注	其他信息
- data_type	string	非必须		数据类型	
format	string	非必须		单位/格式	
- dictionaries	string	非必须		字典	
status	number	非必须		显示状态	
- preset	number	非必须		是否预置	
- position	string	非必须			

# 用户属性

# 用户属性列表

#### 基本信息

Path: /config/userProperty

**Method:** GET

接口描述:

#### 请求参数

#### Query

参数名称	是否必须	示例	备注
_limit[page]	否		页码
_limit[size]	否		每页记录数
name[like]	否		属性名搜索

名称	类型	是否必须	默认值	备注	其他信息	
----	----	------	-----	----	------	--

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object	非必须			
items	object []	非必须			item 类型: object
name	string	非必须		属性名	
show_name	string	非必须		显示名	
is_update	number	非必须		是否可修改	
- data_type	string	非必须		数据类型	
format	string	非必须		单位/格式	
- dictionaries	string	非必须		字典	
status	number	非必须		显示状态	
- preset	number	非必须		预置	
- position	string	非必须		字段位置	
total	number	非必须		总记录数	

# 用户属性编辑

## 基本信息

Path: /config/userProperty

Method: PUT

接口描述:

### 请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### Body

参数名称	参数类型	是否必须	示例	备注
name	text	否		属性名
show_name	text	否		显示名
data_type	text	否		数据类型
format	text	否		单位/格式
dictionaries	text	否		字典
status	text	否		显示状态
remark	text	是		备注

### 返回数据

名称	类型	是否必须	默认值	备注	其他信息	
----	----	------	-----	----	------	--

## 用户属性详情

#### 基本信息

Path: /config/userProperty/getInfo

Method: GET

接口描述:

#### 请求参数

#### Query

参数名称	是否必须	示例	备注
name	否		属性名

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object	非必须			
- name	string	非必须		属性名	
show_name	string	非必须		显示名	
- is_update	number	非必须			
- data_type	string	非必须		数据类型	
- format	string	非必须		单位/格式	
- dictionaries	string	非必须		字典	
- status	number	非必须		显示状态	
- preset	number	非必须		是否预置	
- position	string	非必须			

# 原事件

# 原事件列表

### 基本信息

Path: /config/event

Method: GET

接口描述:

### 请求参数

Query

参数名称	是否必须	示例	备注
_limit[page]	否		页码
_limit[size]	否		每页记录数

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object	非必须			
- items	object []	非必须		记录内容	item 类型: object
name	string	非必须		事件名	
show_name	string	非必须		显示名	
label	string	非必须		标签	
status	number	非必须		显示状态	
remark	string	非必须		备注	
total	number	非必须		总记录数	

# 原事件编辑

## 基本信息

Path: /config/event

Method: PUT

接口描述:

### 请求参数

**Headers** 

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

#### Body

参数名称	参数类型	是否必须	示例	备注
name	text	否		原事件名
show_name	text	否		显示名
label	text	否		标签
status	text	否		显示状态
remark	text	否		备注

### 返回数据

名称	类型	是否必须	默认值	备注	其他信息	

## 原事件详情

### 基本信息

Path: /config/event/getInfo

Method: GET

接口描述:

#### 请求参数

#### Query

参数名称	是否必须	示例	备注
name	是	sdk_push	事件名

名称	类型	是否必须	默认值	备注	其他信息
		. —	*****		

# 设备属性

# 设备属性列表

#### 基本信息

Path: /config/deviceProperty

Method: GET

接口描述:

#### 请求参数

#### Query

参数名称	是否必须	示例	备注
_limit[page]	否		页码
_limit[size]	否		每页记录数
name[like]	否		属性名搜索

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			
msg	string	非必须			
data	object	非必须			
- items	object []	非必须			item 类型: object
name	string	非必须		属性名	

名称	类型	是否必须	默认值	备注	其他信息
show_name	string	非必须		显示名	
├— is_update	number	非必须		是否可修改	
- data_type	string	非必须		数据类型	
- format	string	非必须		单位/格式	
- dictionaries	string	非必须		字典	
status	number	非必须		显示状态	
- preset	number	非必须		预置	
- position	string	非必须		字段位置	
total	number	非必须		总记录数	

# 设备属性编辑

### 基本信息

Path: /config/deviceProperty

Method: PUT

接口描述:

### 请求参数

#### **Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

参数名称	参数类型	是否必须	示例	备注
name	text	否		属性名

参数名称	参数类型	是否必须	示例	备注
show_name	text	否		显示名
data_type	text	否		数据类型
format	text	否		单位/格式
dictionaries	text	否		字典
status	text	否		显示状态
remark	text	是		备注

名称	类型	是否必须	默认值	备注	其他信息	

# 设备属性详情

### 基本信息

Path: /config/deviceProperty/getInfo

Method: GET

接口描述:

#### 请求参数

#### Query

参数名称	是否必须	示例	备注
name	否		属性名

名称	类型	是否必须	默认值	备注	其他信息
status	number	非必须			

名称	类型	是否必须	默认值	备注	其他信息
msg	string	非必须			
data	object	非必须			
- name	string	非必须		属性名	
show_name	string	非必须		显示名	
is_update	number	非必须			
- data_type	string	非必须		数据类型	
├— format	string	非必须		单位/格式	
- dictionaries	string	非必须		字典	
- status	number	非必须		显示状态	
- preset	number	非必须		是否预置	
- position	string	非必须			