

# **Kiloview AVX24-4 Media HUB**

**WEB API 说明文档**

**文档版本： v1.0**

**适用产品： AVX24-4 Media HUB**

**发布单位： 长沙千视电子科技有限公司**

## 文档说明

本文档为 AVX24-4 Media HUB（媒体汇聚 HUB）的 HTTP API 接口说明书，旨在提供完整的后端接口说明，涵盖用户认证、系统管理、网络配置、面板控制、交换机管理及诊断告警等核心功能模块，通过 HTTP 路由定义了各操作的请求方式与路径，服务于系统集成与远程管理。

关键要点包括：

- 用户认证与会话管理

提供登录、登出、用户增删改查及会话验证接口，支持基于令牌的身份认证机制。

- 系统运维与固件升级

包含固件上传、分片更新、升级状态查询、日志获取、重启及恢复出厂设置等系统级操作接口。

- 面板与输入输出控制

支持面板增删改查、输入源发现与管理、输出配置及交叉切换，实现音视频信号的灵活调度。

- 网络与 VLAN 配置管理

提供 VLAN 设置、DHCP 管理、静态路由、PTP 时钟、LLDP 拓扑更新及 KDDs 网络诊断等功能接口。

- 告警事件与交换机深度管控

实现告警查询、配置与导出，以及交换机端口、PoE、聚合、VLAN 和电源状态的精细化管理。

**版本提示：**由于产品持续迭代升级，您所购买的产品可能与本手册的描述存在差异。如有不一致之处，以包装箱内产品实物及随附说明书为准。

## 法律声明

接收长沙千视电子科技有限公司（以下简称"千视电子"）此份文档，即视为您已同意以下条款。若不同意以下条款，请停止使用本文档。

本文档版权归长沙千视电子科技有限公司所有，保留任何未在本文档中明示授予的权利。文档中包含千视电子的专有信息。未经千视电子事先书面许可，任何单位和个人不得复制、传递、分发、使用或泄漏本文档以及本文档包含的任何图片、表格、数据及其他信息。

千视电子的注册商标、产品名称及标志为千视电子所有。本文档中提及的其他产品或公司名称可能是其各自所有者的商标或注册商标。在未经千视电子或第三方权利人事先书面同意的情况下，阅读本文档不表示以默示、不可反言或其他方式授予阅读者使用本文档中出现的任何标记的任何权利。

本产品符合相关环境保护和人身安全方面的设计要求，产品的存放、使用和弃置应遵照产品手册、相关合同或所在国家/地区法律、法规的要求进行。

本文档按"现状"和"仅此状态"提供。本文档中的信息将随千视电子产品和技术的进步而不断更新，千视电子不再单独通知此类信息的更新。

本文档未尽事宜，请访问千视电子官方网站 [www.kiloview.com](http://www.kiloview.com) 获取相关信息和技术支持。

|                                 |    |
|---------------------------------|----|
| 1 用户认证.....                     | 1  |
| 1.1 用户管理 / Users.....           | 1  |
| 1.1.1 用户登录.....                 | 1  |
| 1.1.2 用户删除.....                 | 2  |
| 1.1.3 添加用户.....                 | 3  |
| 1.1.4 修改用户.....                 | 4  |
| 1.1.5 获取所有用户列表.....             | 5  |
| 2 系统管理.....                     | 6  |
| 2.1 系统设置.....                   | 6  |
| 2.1.1 设置系统名称.....               | 6  |
| 2.1.2 恢复出厂设置.....               | 7  |
| 2.1.3 获取系统日志.....               | 8  |
| 2.1.4 重启系统.....                 | 9  |
| 2.1.5 固件上传.....                 | 10 |
| 2.1.6 固件升级.....                 | 11 |
| 2.1.7 获取运行日志.....               | 11 |
| 2.1.8 导出操作日志.....               | 12 |
| 2.2 系统时间.....                   | 13 |
| 2.2.1 获取系统时间时区.....             | 13 |
| 2.2.2 设置时区.....                 | 14 |
| 2.2.3 设置系统时间.....               | 15 |
| 2.3 PTP 时钟.....                 | 17 |
| 2.3.1 查询 PTP 时钟状态.....          | 17 |
| 2.3.2 保存 PTP 时钟启停.....          | 18 |
| 2.3.3 查询 PTP 时钟详情.....          | 19 |
| 2.3.4 修改 PTP 时钟配置.....          | 21 |
| 3 访问策略/预设配置.....                | 23 |
| 3.1. 访问策略.....                  | 23 |
| 3.1.1 获取备份列表.....               | 23 |
| 3.1.2 保存备份.....                 | 24 |
| 3.1.3 读取备份.....                 | 25 |
| 3.1.4 删除备份.....                 | 26 |
| 3.1.5 修改备份名称.....               | 27 |
| 3.2 虚拟网卡接口.....                 | 28 |
| 3.2.1 查询指定 VLAN 的虚拟接口配置.....    | 28 |
| 3.2.2 查询所有 虚拟网卡接口 IP 列表.....    | 29 |
| 3.2.3 设置指定 VLAN 的虚拟网卡接口配置.....  | 30 |
| 3.2.4 查询所有 VLAN 虚拟接口 IP 信息..... | 31 |
| 3.3 模式切换.....                   | 33 |
| 3.3.1 查询交换机二层/三层模式.....         | 33 |

|                          |    |
|--------------------------|----|
| 3.3.2 设置交换机二层/三层模式 ..... | 34 |
| 4 面板管理 .....             | 35 |
| 4.1 输入源发现 .....          | 35 |
| 4.1.1 添加服务器发现 .....      | 35 |
| 4.1.2 添加手动发现 .....       | 36 |
| 4.1.3 添加自动发现 .....       | 37 |
| 4.1.4 服务器发现查询 .....      | 37 |
| 4.1.5 手动发现查询 .....       | 39 |
| 4.1.6 自动发现查询 .....       | 41 |
| 4.1.7 删除发现项 .....        | 42 |
| 4.1.8 获取所有发现项 .....      | 43 |
| 4.2 输入管理 .....           | 45 |
| 4.2.1 修改输入源 .....        | 45 |
| 4.3 输出管理 .....           | 46 |
| 4.3.1 添加输出 .....         | 46 |
| 4.3.2 删除输出 .....         | 47 |
| 4.3.3 修改输出 .....         | 48 |
| 4.3.4 查询输出 .....         | 49 |
| 4.3.5 获取所有输出 .....       | 50 |
| 4.3.6 获取默认输出 .....       | 52 |
| 4.4 面板基础操作 .....         | 53 |
| 4.4.1 添加面板 .....         | 53 |
| 4.4.2 删除面板 .....         | 54 |
| 4.4.3 修改面板 .....         | 55 |
| 4.4.4 展开面板 .....         | 56 |
| 4.4.5 查询面板 .....         | 57 |
| 4.4.6 获取所有面板 .....       | 58 |
| 4.5 面板输入管理 .....         | 59 |
| 4.5.1 添加面板输入 .....       | 59 |
| 4.5.2 移除面板输入 .....       | 61 |
| 4.5.3 面板输入排序 .....       | 62 |
| 4.6 面板输出管理 .....         | 63 |
| 4.6.1 添加面板输出 .....       | 63 |
| 4.6.2 移除面板输出 .....       | 64 |
| 4.6.3 面板输出排序 .....       | 65 |
| 4.7 面板交叉切换 .....         | 66 |
| 4.7.1 获取面板详情 .....       | 66 |
| 4.7.2 输出交叉输入 .....       | 69 |
| 4.7.3 输入选择 .....         | 70 |
| 4.8 面板配置 .....           | 71 |
| 4.8.1 获取所有输入配置 .....     | 71 |

|                         |     |
|-------------------------|-----|
| 4.8.2 获取所有输出配置 .....    | 73  |
| 4.8.3 应用输入配置 .....      | 74  |
| 4.8.4 应用输出配置 .....      | 75  |
| 4.8.5 设置物理面板启用状态 .....  | 76  |
| 4.8.6 查询物理面板启用状态 .....  | 77  |
| 5 KLS 面板管理 .....        | 78  |
| 5.1 KLS 输入源发现管理 .....   | 78  |
| 5.1.1 KLS 添加服务器发现 ..... | 78  |
| 5.1.2 KLS 添加手动发现 .....  | 79  |
| 5.1.3 KLS 添加自动发现 .....  | 80  |
| 5.1.4 KLS 删除发现项 .....   | 81  |
| 5.2 KLS 输出管理 .....      | 82  |
| 5.2.1 KLS 添加输出 .....    | 82  |
| 5.2.2 KLS 删除输出 .....    | 84  |
| 5.2.3 KLS 修改输出 .....    | 85  |
| 5.2.4 KLS 查询输出 .....    | 86  |
| 5.2.5 KLS 获取所有输出 .....  | 87  |
| 5.3 KLS 面板基础操作 .....    | 88  |
| 5.3.1 KLS 添加面板 .....    | 88  |
| 5.3.2 KLS 删除面板 .....    | 89  |
| 5.3.3 KLS 交换机已启用 .....  | 90  |
| 5.3.4 KLS 删除设备 .....    | 94  |
| 5.4 KLS 面板输入管理 .....    | 95  |
| 5.4.1 KLS 添加面板输入 .....  | 95  |
| 5.4.2 KLS 移除面板输入 .....  | 96  |
| 5.5 KLS 面板输出管理 .....    | 97  |
| 5.5.1 KLS 添加面板输出 .....  | 97  |
| 5.5.2 KLS 移除面板输出 .....  | 98  |
| 5.6 KLS 面板交叉切换 .....    | 99  |
| 5.6.1 KLS 获取面板详情 .....  | 99  |
| 5.6.2 KLS 输出交叉输入 .....  | 102 |
| 5.6.3 KLS 输入全选 .....    | 103 |
| 6 KLS 绑定管理 .....        | 104 |
| 6.1 绑定 KLS .....        | 104 |
| 6.2 解绑 KLS .....        | 105 |
| 请求参数说明 .....            | 105 |
| 6.3 查询 KLS VLAN .....   | 106 |
| 6.4 获取绑定信息 .....        | 107 |
| 6.5 获取绑定状态 .....        | 108 |
| 7 图标管理 .....            | 109 |
| 7.1 上传图标 .....          | 109 |

|                             |     |
|-----------------------------|-----|
| 7.2 修改图标 .....              | 110 |
| 7.3 获取所有图标 .....            | 111 |
| 7.4 通知图标更新 .....            | 112 |
| 8 前面板操作 .....               | 113 |
| 8.1 输入预览 .....              | 113 |
| 8.2 执行切换 .....              | 114 |
| 8.3 查询前面板 .....             | 115 |
| 8.4 获取前面板详情 .....           | 116 |
| 8.5 停止 NDI .....            | 118 |
| 9 VLAN 配置 .....             | 119 |
| 9.1 获取所有 VLAN 配置 .....      | 119 |
| 9.2 修改 VLAN 配置 .....        | 120 |
| 10 拓扑管理 .....               | 121 |
| 10.1 查询拓扑结构 .....           | 121 |
| 10.2 查询拓扑状态 .....           | 124 |
| 10.3 查询设备信息 .....           | 128 |
| 11 网络诊断 .....               | 131 |
| 11.1 诊断管理 .....             | 131 |
| 11.1.1 KLS 跨设备开启端到端诊断 ..... | 131 |
| 11.1.2 获取当前交换机所有设备列表 .....  | 133 |
| 11.1.3 端诊断任务删除 .....        | 134 |
| 11.1.4 诊断设备结果查询 .....       | 135 |
| 11.1.5 开启端到端诊断 .....        | 139 |
| 11.1.6 查询诊断任务（按类型） .....    | 140 |
| 12 交换机管理 .....              | 142 |
| 12.1 系统信息 .....             | 142 |
| 12.1.1 查询交换机系统信息 .....      | 142 |
| 12.1.2 设置设备别名 .....         | 143 |
| 12.2 端口管理 .....             | 144 |
| 12.2.1 查询所有端口状态 .....       | 144 |
| 12.2.2 端口控制 .....           | 149 |
| 12.2.3 设置端口 .....           | 150 |
| 12.2.4 批量设置端口 .....         | 151 |
| 12.2.5 设置端口别名 .....         | 153 |
| 12.2.6 查询端口流量 .....         | 154 |
| 12.2.7 查询端口流量 top .....     | 156 |
| 12.2.8 设置关注端口 .....         | 158 |
| 12.2.9 删除关注端口 .....         | 159 |
| 12.2.10 查询关注端口信息 .....      | 160 |
| 12.2.11 移除设备 .....          | 163 |
| 12.3 端口聚合 .....             | 163 |

|                                           |     |
|-------------------------------------------|-----|
| 12.3.1 创建链路聚合组 .....                      | 163 |
| 12.3.2 删除链路聚合组 .....                      | 165 |
| 12.3.3 查询全部链路聚合组 .....                    | 166 |
| 12.3.4 查询指定或全部链路聚合组 .....                 | 167 |
| 12.3.5 向聚合组追加端口 .....                     | 168 |
| 12.3.6 修改链路聚合组配置 .....                    | 169 |
| 12.4 VLAN 管理 .....                        | 170 |
| 12.4.1 查询所有 VLAN 信息 .....                 | 170 |
| 12.4.2 添加 VLAN .....                      | 173 |
| 12.4.3 设置 VLAN .....                      | 175 |
| 12.4.4 查询 VLAN 信息 .....                   | 176 |
| 12.4.5 删除 VLAN .....                      | 177 |
| 12.4.6 设置 VLAN 端口 .....                   | 178 |
| 12.5 DHCP .....                           | 179 |
| 12.5.1 查询 DHCP 客户端列表 .....                | 179 |
| 12.5.2 获取指定 VLAN 的 VLAN 接口与 DHCP 配置 ..... | 181 |
| 12.5.3 设置指定 VLAN 的 VLAN 接口与 DHCP 配置 ..... | 183 |
| 12.5.4 开启或关闭指定 VLAN 的 DHCP 服务 .....       | 184 |
| 12.5.5 获取静态 IP 绑定列表 .....                 | 185 |
| 12.5.6 批量设置静态 IP 绑定 .....                 | 186 |
| 12.6 电源管理 .....                           | 188 |
| 12.6.1 查询电源信息 .....                       | 188 |
| 12.7 静态路由 .....                           | 189 |
| 12.7.1 添加静态路由 .....                       | 189 |
| 12.7.2 删除静态路由 .....                       | 190 |
| 12.7.3 获取全部静态路由列表 .....                   | 191 |
| 12.8 网络保存 .....                           | 192 |
| 12.8.1 保存网络配置 .....                       | 192 |
| 12.8.2 查询网络配置 .....                       | 193 |
| 13 告警事件 .....                             | 194 |
| 13.1 告警事件管理 .....                         | 194 |
| 13.1.1 按设备查询告警事件 .....                    | 194 |
| 13.1.2 删除告警事件 .....                       | 195 |
| 13.1.3 分页查询告警事件 .....                     | 196 |
| 13.1.4 按告警级别筛选事件列表 .....                  | 198 |
| 13.1.5 批量设置告警事件状态 .....                   | 200 |
| 13.1.6 下载告警导出文件 .....                     | 201 |
| 13.1.7 导出告警列表并获取文件路径 .....                | 201 |
| 13.2 告警配置 .....                           | 203 |
| 13.2.1 获取告警策略配置 .....                     | 203 |
| 13.2.2 设置告警策略配置 .....                     | 204 |

|                                 |     |
|---------------------------------|-----|
| 13.3 告警调试 .....                 | 205 |
| 13.3.1 调试：查询全部端口告警（DEBUG） ..... | 205 |
| 13.3.2 调试：查询端口告警（DEBUG） .....   | 207 |
| 14 生产测试 .....                   | 208 |
| 14.1 生产测试与设备信息 .....            | 208 |
| 14.1.1 恢复出厂设置 .....             | 208 |
| 14.1.2 获取风扇列表 .....             | 209 |
| 14.1.3 获取设备资源与版本信息 .....        | 210 |
| 14.1.4. 查询交换机端口状态 .....         | 212 |
| 14.1.5 设置序列号 .....              | 213 |
| 14.1.6 启动风扇测试 .....             | 214 |
| 14.1.7 停止风扇测试 .....             | 215 |
| 14.1.8 查询 USB 端口状态 .....        | 216 |
| 15 AVX 页面接口 .....               | 217 |
| 15.1 获取系统信息 .....               | 217 |
| 15.2 获取网络接口 .....               | 218 |
| 16 AVX 反代设备页面接口 .....           | 220 |
| 16.1 获取子设备 iframe 页面接口 .....    | 220 |

## 1 用户认证

## 1.1 用户管理 / Users

### 1.1.1 用户登录

## 接口描述

登录验证，登录成功后签发接口鉴权用 token。

**请求地址**

/v1/user/login

## 请求方法

POST

### 请求参数示例

```
1 {
2     "username": "admin",
3     "password": "admin"
4 }
```

## 请求参数说明

表格

| 参数名称     | 参数类型   | 是否必填 | 描述     |
|----------|--------|------|--------|
| username | string | True | 用户登录账号 |
| password | string | True | 用户登录密码 |

## 响应示例

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {  
5     "alias": "admin",  
6     "token":  
7       "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwiaWF0IjoxNjcjbM  
8         iOjtZWRRPS1zd2l0Y2hlciIsIm5iZiI6MTc0OTYyNzY1MH0.Mjs7dXpXkSbnPczye17visjuK-RwZEMoHoddwH9YnQ"  
9   }  
}
```

## 响应参数说明

| 参数名称       | 参数类型   | 描述                   |
|------------|--------|----------------------|
| code       | int32  | 业务状态码，200 表示成功       |
| message    | string | 业务提示信息               |
| data       | object | 登录返回核心数据             |
| data.alias | string | 用户设置的别名 / 昵称         |
| data.token | string | 登录成功后签发的接口鉴权 Token 值 |

## 1.1.2 用户删除

### 接口描述

删除指定系统用户，支持批量删除。

### 请求地址

/v1/user/delete

### 请求方法

POST

### 请求参数示例

```
1 {
2   "ids": [2, 3, 4]
3 }
```

### 请求参数说明

| 参数名称 | 参数类型  | 是否必填 | 描述           |
|------|-------|------|--------------|
| ids  | array | True | 待删除的用户 ID 数组 |

### 响应示例

```
1 {
2   "code": 200,
3   "message": "用户删除成功",
4   "data": null
5 }
```

### 应参数说明

| 参数名称 | 参数类型 | 描述 |
|------|------|----|
|------|------|----|

|         |        |                   |
|---------|--------|-------------------|
| code    | int32  | 业务状态码，200 表示成功    |
| message | string | 操作结果描述            |
| data    | null   | 返回数据，无业务数据时为 null |

### 1.1.3 添加用户

#### 接口描述

新增系统可登录用户，支持设置管理员 / 普通用户身份。

#### 请求地址

/v1/user/add

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "alias": "csadmin",
3   "password": "123",
4   "role": "admin",
5   "username": "csadmin"
6 }
```

#### 请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述                                        |
|----------|--------|------|-------------------------------------------|
| alias    | string | True | 用户昵称 / 别名                                 |
| password | string | True | 用户登录密码                                    |
| role     | string | True | 用户身份。枚举：<br>common - 普通用户，<br>admin - 管理员 |
| username | string | True | 用户登录账号                                    |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "用户创建成功",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述                |
|---------|--------|-------------------|
| code    | int32  | 业务状态码，200 表示成功    |
| message | string | 操作结果描述            |
| data    | null   | 返回数据，无业务数据时为 null |

1.1.4 修改用户

接口描述

修改用户的昵称、密码、角色等基础信息。

请求地址

/v1/user/modify

请求方法

POST

请求参数示例

```
1 {
2   "alias": "csadmin_new",
3   "id": 0,
4   "password": "newpassword123",
5   "role": "common"
6 }
```

请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述                                            |
|----------|--------|------|-----------------------------------------------|
| alias    | string | True | 修改后的用户昵称 / 别名                                 |
| id       | string | True | 需要修改的目标用户 ID                                  |
| password | string | True | 用户修改后的新密码                                     |
| role     | string | True | 修改后的用户身份。枚举：<br>common - 普通用户，<br>admin - 管理员 |

响应示例

```
1 {
2   "code": 200,
3   "message": "修改用户信息成功",
4   "data": null
5 }
```

## 响应参数说明

| 参数名称    | 参数类型   | 描述                |
|---------|--------|-------------------|
| code    | int32  | 业务状态码，200 表示成功    |
| message | string | 操作结果描述            |
| data    | null   | 返回数据，无业务数据时为 null |

## 1.1.5 获取所有用户列表

### 接口描述

获取当前系统已注册的全部用户列表与详细信息。

### 请求地址

/v1/user/userlist

### 请求方法

GET

### 请求参数示例

无

### 请求参数说明

无

### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "id": 1,
7       "username": "admin",
8       "alias": "admin",
9       "role": "admin",
10      "online": true,
11      "first_login": true,
12      "create_time": "2026-02-04 18:51:27",
13      "update_time": "2026-02-04 18:51:27"
14    }
15  ]
16 }
```

## 响应参数说明

| 参数名称             | 参数类型   | 描述                                |
|------------------|--------|-----------------------------------|
| code             | int32  | 业务状态码，200 表示成功                    |
| message          | string | 业务提示信息                            |
| data             | array  | 系统用户列表数组                          |
| data.id          | int32  | 用户 ID                             |
| data.username    | string | 用户登录账号                            |
| data.alias       | string | 用户昵称 / 别名                         |
| data.role        | string | 用户身份。枚举：common - 普通用户，admin - 管理员 |
| data.online      | bool   | 用户当前是否在线                          |
| data.first_login | bool   | 用户是否为首次登录                         |
| data.create_time | string | 用户创建时间，格式：yyyy-MM-dd HH:mm:ss     |
| data.update_time | string | 用户信息最后更新时间，格式：yyyy-MM-dd HH:mm:ss |

## 2 系统管理

### 2.1 系统设置

#### 2.1.1 设置系统名称

##### 接口描述

修改设备的系统主机名称。

##### 请求地址

/v1/sys/sethostname

##### 请求方法

GET

##### 请求参数示例

请求地址示例：/v1/sys/sethostname?hostname=AVX-Core-Switch-001

请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述              |
|----------|--------|------|-----------------|
| hostname | string | True | 系统主机名称，Query 参数 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {}
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

2.1.2 恢复出厂设置

接口描述

将设备所有配置恢复至出厂默认状态。

请求地址

/v1/sys/refactory

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
```

```
2     "code": 200,
3     "message": "ok",
4     "data": {}
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

2.1.3 获取系统日志

接口描述

获取设备运行的系统日志文本内容。

请求地址

/v1/sys/syslog

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": "2024-01-01 00:00:00 system boot success\n2024-01-01 00:00:05 network interface
up\n2024-01-01 10:00:00 diagnose task 2001 start\n2024-01-01 10:03:00 diagnose task 2001 finished"
5 }
```

响应参数说明

| 参数名称 | 参数类型  | 描述             |
|------|-------|----------------|
| code | int32 | 业务状态码，200 表示成功 |

|         |        |            |
|---------|--------|------------|
| message | string | 业务提示信息     |
| data    | string | 系统日志完整文本内容 |

## 2.1.4 重启系统

### 接口描述

执行设备系统软重启操作。

### 请求地址

/v1/sys/reboot

### 请求方法

GET

### 请求参数示例

无

### 请求参数说明

无

### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {}
5 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

## 2.1.5 固件上传

### 接口描述

上传固件文件，采用 JSON 格式提交，文件内容以 base64 编码传输。

### 请求地址

/v1/sys/upload

### 请求方法

POST

### 请求参数示例

```
1 {  
2   "u_file": "base64-encoded-bytes",  
3   "filename": "AVX-Firmware.bin"  
4 }
```

### 请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述                        |
|----------|--------|------|---------------------------|
| u_file   | string | True | 上传的固件文件内容，<br>base64 编码格式 |
| filename | string | True | 固件文件名称                    |

### 响应示例

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

### 响应参数说明

表格

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据           |

## 2.1.6 固件升级

### 接口描述

触发设备固件升级操作，需先通过固件上传接口完成升级包上传。

### 请求地址

/v1/sys/upgrade

### 请求方法

POST

### 请求参数示例

```
1 {  
2   "filename": "AVX-Firmware.bin"  
3 }
```

### 请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述        |
|----------|--------|------|-----------|
| filename | string | True | 固件升级包文件名称 |

### 响应示例

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

## 2.1.7 获取运行日志

### 接口描述

获取设备系统运行日志文本内容。

请求地址

/v1/sys/getlog

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": "log content..."
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | string | 系统运行日志完整文本内容   |

2.1.8 导出操作日志

接口描述

将系统操作日志打包为 txt 文件，返回文件存储路径用于后续下载。

请求地址

/v1/sys/operation/export

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": "/data/operation/operation.txt"
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | string | 操作日志导出后的文件存储路径 |

2.2 系统时间

2.2.1 获取系统时间时区

接口描述

获取设备当前的系统时间、时区、NTP 配置等相关信息。

请求地址

/v1/systime/get

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2     "code": 200,
3     "message": "",
4     "data": {
5         "Timezone": "Asia/Shanghai",
6         "ntp": "time.windows.com time.apple.com pool.ntp.org",
7         "offset": 8,
8         "time": "2026-03-18 15:54:48",
9         "timetype": "pc"
10    }
11 }
```

响应参数说明

| 参数名称          | 参数类型   | 描述                                                        |
|---------------|--------|-----------------------------------------------------------|
| code          | int32  | 业务状态码，200 表示成功                                            |
| message       | string | 业务提示信息                                                    |
| data          | object | 系统时间时区核心数据                                                |
| data.Timezone | string | 系统当前时区                                                    |
| data.ntp      | string | 系统配置的 NTP 服务器地址                                           |
| data.offset   | int32  | 系统时差                                                      |
| data.time     | string | 系统当前时间，格式：<br>yyyy-MM-dd HH:mm:ss                         |
| data.timetype | string | 系统时间类型。枚举：pc - 与 PC<br>同步， manual - 手动设置，<br>ntp-NTP 时间同步 |

2.2.2 设置时区

接口描述

设置系统的时区与时差。

请求地址

/v1/systime/setLocation

请求方法

POST

请求参数示例

```
1 {
2     "timezone": "Asia/Shanghai",
3     "offset": 8
4 }
```

请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述        |
|----------|--------|------|-----------|
| timezone | string | True | 目标时区      |
| offset   | int32  | True | 目标时区对应的时差 |

响应示例

```
1 {
2     "code": 200,
3     "message": "设置地区时间成功",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述                |
|---------|--------|-------------------|
| code    | int32  | 业务状态码，200 表示成功    |
| message | string | 操作结果描述            |
| data    | null   | 返回数据，无业务数据时为 null |

2.2.3 设置系统时间

接口描述

设置系统时间类型、手动时间、NTP 同步地址等时间相关配置。

请求地址

/v1/systime/setLocation

请求方法

POST

请求参数示例

NTP 时间同步配置示例：

```
1 {
2     "timetype": "ntp",
```

```

3   "offset": 8,
4   "ntp": "time.windows.com time.apple.com pool.ntp.org"
5 }

```

手动设置时间示例:

```

1 {
2   "timetype": "manual",
3   "offset": 8,
4   "time": "2026-04-16 15:07:44"
5 }

```

与 PC 时间同步示例:

```

1 {
2   "timetype": "pc",
3   "offset": 8
4 }

```

### 请求参数说明

| 参数名称     | 参数类型   | 是否必填  | 描述                                                        |
|----------|--------|-------|-----------------------------------------------------------|
| timetype | string | True  | 时间类型。枚举: manual - 手动设置, ntp-NTP 时间同步, pc - 与 PC 时间一致      |
| offset   | int32  | True  | 系统时差                                                      |
| time     | string | False | 手动设置的系统时间, 格式 : yyyy-MM-dd HH:mm:ss ; timetype 为 ntp 时非必填 |
| ntp      | string | False | NTP 时间同步服务器地址 ; 仅 timetype 为 ntp 时需配置                     |

### 响应示例

```

1 {
2   "code": 200,
3   "message": "设置时间成功",
4   "data": null
5 }

```

### 响应参数说明

| 参数名称    | 参数类型   | 描述                 |
|---------|--------|--------------------|
| code    | int32  | 业务状态码, 200 表示成功    |
| message | string | 操作结果描述             |
| data    | null   | 返回数据, 无业务数据时为 null |

## 2.3 PTP 时钟

### 2.3.1 查询 PTP 时钟状态

#### 接口描述

查询 PTP 时钟状态与端口绑定信息。

#### 请求地址

/v1/ptp/query

#### 请求方法

GET

#### 请求参数示例

无

#### 请求参数说明

无

#### 响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {
5         "ports": [],
6         "ptps": []
7     }
8 }
```

#### 响应参数说明

| 参数名称              | 参数类型   | 描述                    |
|-------------------|--------|-----------------------|
| code              | int32  | 业务状态码，200 表示成功        |
| message           | string | 业务提示信息                |
| data              | object | 返回核心数据                |
| data.ports        | array  | 端口列表（SWPort）          |
| data.ports.id     | int32  | 端口 ID                 |
| data.ports.ptp_id | int32  | 绑定的 PTP 时钟 ID，0 表示未绑定 |

|                                 |        |                                      |
|---------------------------------|--------|--------------------------------------|
| data.ports.mode                 | string | 端口模式。枚举：e2e/p2p                      |
| data.ports.master               | bool   | 是否为主端口                               |
| data.ports.vlanInfo             | object | 端口 VLAN 信息                           |
| data.ports.vlanInfo.type        | string | VLAN 模式                              |
| data.ports.vlanInfo.pvid        | int32  | VLAN PVID                            |
| data.ports.vlanInfo.vid         | string | VLAN 范围                              |
| data.ports.vlanInfo.alias       | string | VLAN 别名                              |
| data.ports.vlanInfo.color       | string | VLAN 颜色                              |
| data.ports.vlanInfo.config_file | string | VLAN 配置文件                            |
| data.ptps                       | array  | PTP 时钟实例列表 (SWPtp)                   |
| data.ptps.id                    | int32  | 时钟实例 ID                              |
| data.ptps.enable                | bool   | 时钟是否启用                               |
| data.ptps.domain                | int32  | 时钟域                                  |
| data.ptps.type                  | string | 时钟类型。枚举：BC/TC/stop                   |
| data.ptps.identity              | string | 时钟标识                                 |
| data.ptps.pport_id              | string | 父端口 ID                               |
| data.ptps.status                | int32  | 时钟运行状态。枚举：0 - 停止，<br>1 - 运行，3 - 配置错误 |
| data.ptps.two_step              | bool   | 是否开启两步模式                             |
| data.ptps.priority1             | int32  | 时钟优先级 1                              |
| data.ptps.priority2             | int32  | 时钟优先级 2                              |
| data.ptps.leap_s                | bool   | 闰秒开关                                 |
| data.ptps.de_mainclock          | string | 与主时钟的偏差值                             |
| data.ptps.average_delay         | string | 平均路径延迟                               |
| data.ptps.upstream_equipment    | string | 上游设备名称                               |
| data.ptps.upstream_port         | string | 上游端口 ID                              |
| data.ptps.ports                 | array  | 时钟绑定的端口列表，结构同<br>data.ports          |

## 2.3.2 保存 PTP 时钟启停

### 接口描述

保存 PTP 时钟启用状态，支持批量配置多个时钟实例的启停。

### 请求地址

/v1/ptp/save

请求方法

POST

请求参数示例

```
1 {
2     "ptps": [
3         {
4             "id": 1,
5             "enable": true
6         },
7         {
8             "id": 2,
9             "enable": false
10        }
11    ]
12 }
```

请求参数说明

| 参数名称        | 参数类型  | 是否必填 | 描述               |
|-------------|-------|------|------------------|
| ptps        | array | True | 时钟启停配置列表         |
| ptps.id     | int32 | True | 时钟实例 ID，取值范围 1-3 |
| ptps.enable | bool  | True | 是否启用该时钟实例        |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {}
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

2.3.3 查询 PTP 时钟详情

接口描述

查询指定 PTP 时钟实例的详细配置信息。

请求地址

/v1/ptp/detail/query

请求方法

GET

请求参数示例

请求地址示例：/v1/ptp/detail/query?id=1

请求参数说明

| 参数名称 | 参数类型  | 是否必填 | 描述                        |
|------|-------|------|---------------------------|
| id   | int32 | True | 时钟实例 ID，取值范围 1-3，Query 参数 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "id": 1,
6     "enable": true,
7     "domain": 0,
8     "type": "BC",
9     "two_step": true,
10    "priority1": 128,
11    "priority2": 128,
12    "leap_s": false,
13    "ports": []
14  }
15 }
```

响应参数说明

| 参数名称           | 参数类型   | 描述                 |
|----------------|--------|--------------------|
| code           | int32  | 业务状态码，200 表示成功     |
| message        | string | 业务提示信息             |
| data           | object | 时钟详情核心数据           |
| data.id        | int32  | 时钟实例 ID            |
| data.enable    | bool   | 时钟是否启用             |
| data.domain    | int32  | 时钟域                |
| data.type      | string | 时钟类型。枚举：BC/TC/stop |
| data.two_step  | bool   | 是否开启两步模式           |
| data.priority1 | int32  | 时钟优先级 1，取值范围 0-255 |

|                     |        |                           |
|---------------------|--------|---------------------------|
| data.priority2      | int32  | 时钟优先级 2，取值范围 0-255        |
| data.leap_s         | bool   | 闰秒开关                      |
| data.ports          | array  | 时钟绑定的端口列表（SWPort）         |
| data.ports.id       | int32  | 端口 ID                     |
| data.ports.ptp_id   | int32  | 绑定的 PTP 时钟 ID             |
| data.ports.mode     | string | 端口模式。枚举：e2e/p2p           |
| data.ports.master   | bool   | 是否为主端口                    |
| data.ports.vlanInfo | object | 端口 VLAN 信息，结构同 4.2.1 接口定义 |

### 2.3.4 修改 PTP 时钟配置

#### 接口描述

修改指定 PTP 时钟实例的基础配置与端口绑定规则。

#### 请求地址

/v1/ptp/modify

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "id": 1,
3     "type": "BC",
4     "domain": 0,
5     "two_step": true,
6     "priority1": 128,
7     "priority2": 128,
8     "leap_s": false,
9     "ports": [
10        {
11            "id": 1,
12            "mode": "e2e",
13            "master": true
14        }
15    ]
16 }
```

#### 请求参数说明

| 参数名称 | 参数类型  | 是否必填 | 描述           |
|------|-------|------|--------------|
| id   | int32 | True | 时钟实例 ID，取值范围 |

|              |        |       |                            |
|--------------|--------|-------|----------------------------|
|              |        |       | 1-3                        |
| type         | string | True  | 时钟类型。枚举：<br>BC/TC/stop     |
| domain       | int32  | False | 时钟域，不传默认值为<br>id-1         |
| two_step     | bool   | False | 是否开启两步模式                   |
| priority1    | int32  | False | 时钟优先级 1，取值范<br>围 0-255     |
| priority2    | int32  | False | 时钟优先级 2，取值范<br>围 0-255     |
| leap_s       | bool   | False | 闰秒开关                       |
| ports        | array  | False | 端口配置列表                     |
| ports.id     | int32  | True  | 端口 ID                      |
| ports.ptp_id | int32  | False | 绑定的 PTP 时钟 ID              |
| ports.mode   | string | True  | 端口模式。枚举：<br>e2e/p2p        |
| ports.master | bool   | False | 是否为主端口，TC 模式<br>将强制为 false |

## 响应示例

```
1 {  
2     "code": 200,  
3     "message": "ok",  
4     "data": {}  
5 }
```

## 响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

## 3 访问策略/预设配置

### 3.1. 访问策略

#### 3.1.1 获取备份列表

##### 接口描述

获取系统中所有已保存的配置预设列表。

##### 请求地址

/v1/preset/getlist

##### 请求方法

GET

##### 请求参数示例

无

##### 请求参数说明

无

##### 响应示例

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": [  
5     {  
6       "prname": "default_config",  
7       "version": "1.0.0",  
8       "apply": true  
9     },  
10    {  
11      "prname": "performance_mode",  
12      "version": "1.2.0",  
13      "apply": false  
14    }  
15  ]  
16 }
```

响应参数说明

| 参数名称           | 参数类型   | 描述             |
|----------------|--------|----------------|
| code           | int32  | 业务状态码，200 表示成功 |
| message        | string | 业务提示信息         |
| data           | array  | 预设列表           |
| data[].prname  | string | 预设名称           |
| data[].version | string | 预设版本号          |
| data[].apply   | bool   | 预设是否可用         |

3.1.2 保存备份

接口描述

将设备当前运行配置保存为新的预设模板。

请求地址

/v1/preset/save

请求方法

POST

请求参数示例

```
1 {
2   "prname": "high_performance_v2"
3 }
```

请求参数说明

| 参数名称   | 参数类型   | 是否必填 | 描述                |
|--------|--------|------|-------------------|
| prname | string | True | 预设名称，不允许包含 "-" 字符 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {}
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

3.1.3 读取备份

接口描述

读取指定预设并将其配置应用到当前设备。

请求地址

/v1/preset/read

请求方法

POST

请求参数示例

```
1 {
2   "prname": "default_config"
3 }
```

请求参数说明

| 参数名称   | 参数类型   | 是否必填 | 描述   |
|--------|--------|------|------|
| prname | string | True | 预设名称 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {}
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

### 3.1.4 删除备份

#### 接口描述

删除指定预设，支持批量删除。

#### 请求地址

/v1/preset/delete

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "namelist": ["old_config", "test_mode_v1", "invalid_preset"]
3 }
```

#### 请求参数说明

| 参数名称     | 参数类型  | 是否必填 | 描述     |
|----------|-------|------|--------|
| namelist | array | True | 预设名称数组 |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "delete_info": {
5     "delete_list": ["old_config", "test_mode_v1"],
6     "notfound": ["invalid_preset"],
7     "failed": [],
8     "message": "2 个预设删除成功, 1 个预设未找到"
9   }
10 }
```

#### 响应参数说明

| 参数名称                    | 参数类型   | 描述             |
|-------------------------|--------|----------------|
| code                    | int32  | 业务状态码，200 表示成功 |
| message                 | string | 业务提示信息         |
| delete_info             | object | 删除结果详情         |
| delete_info.delete_list | array  | 删除成功的预设名称列表    |

|                      |        |             |
|----------------------|--------|-------------|
| delete_info.notfound | array  | 未找到的预设名称列表  |
| delete_info.failed   | array  | 删除失败的预设名称列表 |
| delete_info.message  | string | 删除结果文本描述    |

### 3.1.5 修改备份名称

#### 接口描述

修改已存在预设的名称。

#### 请求地址

/v1/preset/modify

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "old_name": "performance_mode",
3   "new_name": "high_performance_v1"
4 }
```

#### 请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述    |
|----------|--------|------|-------|
| old_name | string | True | 原预设名称 |
| new_name | string | True | 新预设名称 |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {}
5 }
```

#### 响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

## 3.2 虚拟网卡接口

### 3.2.1 查询指定 VLAN 的虚拟接口配置

#### 接口描述

查询指定 VLAN 的虚拟接口配置信息

#### 请求地址

/v1/preset/vlan/interface/get

#### 请求方法

GET

#### 请求参数示例

```
1 {}
```

#### 请求参数说明

| 参数名称    | 参数类型  | 是否必填 | 描述                  |
|---------|-------|------|---------------------|
| vlan_id | int32 | True | VLAN ID (位置: query) |

#### 响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {
5         "interface": {
6             "enable": true,
7             "ip": "192.168.1.1",
8             "netmask": "255.255.255.0",
9             "type": "static",
10            "referenced": true,
11            "is_default_route": true
12        },
13        "recommend": {
14            "dhcp": {
15                "is_recommend": false,
16                "recommend_type": "dhcp"
17            },
18            "manual": {
19                "is_recommend": true,
20                "recommend_type": "manual",
21                "vlanif_ip": "192.168.1.254",
22                "recommend_range": "192.168.1.1-192.168.1.200"
23            }
24        }
25    }
26 }
```

```
25     }  
26 }
```

响应参数说明

| 参数名称                             | 参数类型   | 描述                                  |
|----------------------------------|--------|-------------------------------------|
| code                             | int32  | 业务状态码，200 表示成功                      |
| message                          | string | 业务提示信息                              |
| interface                        | object | 虚拟接口配置信息                            |
| interface.enable                 | bool   | 虚拟接口使能开关                            |
| interface.ip                     | string | 虚拟接口 IP 地址                          |
| interface.netmask                | string | 子网掩码                                |
| interface.type                   | string | 接口类型                                |
| interface.referenced             | bool   | 是否被矩阵引用                             |
| interface.is_default_route       | bool   | 是否设置为默认路由                           |
| recommend                        | object | 推荐配置信息 (DHCP/手动配置)<br>该字段仅为前端推荐提示使用 |
| recommend.dhcp                   | object | 业务参数                                |
| recommend.dhcp.is_recommend      | bool   | 布尔开关值                               |
| recommend.dhcp.recommend_type    | string | 返回的类型标识                             |
| recommend.manual                 | object | 业务参数                                |
| recommend.manual.is_recommend    | bool   | 布尔开关值                               |
| recommend.manual.recommend_type  | string | 返回的类型标识                             |
| recommend.manual.vlanif_ip       | string | 返回的 IP 地址                           |
| recommend.manual.recommend_range | string | 字符串参数                               |

- GET /v1/preset/vlan/interface/list/get - 获取虚拟网卡接口列表
- POST /v1/preset/vlan/interface/set - 设置虚拟网卡接口

3.2.2 查询所有 虚拟网卡接口 IP 列表

接口描述

查询所有虚拟网卡的 IP 配置列表（带别名和颜色信息）

请求地址

/v1/preset/vlan/interface/list/get

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": [
5         {
6             "vlan_id": 1,
7             "ip": "string",
8             "netmask": "string",
9             "alias": "string",
10            "color": "string"
11        }
12    ]
13 }
```

响应参数说明

| 参数名称         | 参数类型   | 描述              |
|--------------|--------|-----------------|
| code         | int32  | 业务状态码，200 表示成功  |
| message      | string | 业务提示信息          |
| data         | array  | VLAN 接口 IP 配置列表 |
| data.vlan_id | int32  | VLAN ID         |
| data.ip      | string | IP 地址           |
| data.netmask | string | 子网掩码            |
| data.alias   | string | VLAN 别名         |
| data.color   | string | VLAN 颜色         |

3.2.3 设置指定 VLAN 的虚拟网卡接口配置

接口描述

设置指定 VLAN 的虚拟网卡配置

请求地址

/v1/preset/vlan/interface/set

请求方法

POST

请求参数示例

```
1 {
2   "vlan_id": 10,
3   "interface": {
4     "enable": true,
5     "ip": "192.168.10.1",
6     "netmask": "255.255.255.0",
7     "type": "static",
8     "is_default_route": false
9   }
10 }
```

请求参数说明

| 参数名称                       | 参数类型   | 是否必填  | 描述         |
|----------------------------|--------|-------|------------|
| vlan_id                    | int32  | True  | VLAN ID    |
| interface                  | object | True  | 虚拟接口配置信息   |
| interface.enable           | bool   | True  | 虚拟接口使能开关   |
| interface.ip               | string | True  | 虚拟接口 IP 地址 |
| interface.netmask          | string | True  | 子网掩码       |
| interface.type             | string | False | 接口类型       |
| interface.is_default_route | bool   | False | 是否设置为默认路由  |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok"
4 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

3.2.4 查询所有 VLAN 虚拟接口 IP 信息

接口描述

查询所有 VLANIF 的 IP 地址信息  
给小屏提供获取所有虚拟网卡 IP 接口

请求地址

/v1/switch/interface/ip/get

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {
5         "interface_ip_list": [
6             {
7                 "vlan_id": 0,
8                 "ip": "192.168.0.168"
9             },
10            {
11                "vlan_id": 1,
12                "ip": "192.168.1.1"
13            }
14        ]
15    }
16 }
```

响应参数说明

| 参数名称                           | 参数类型   | 描述                                |
|--------------------------------|--------|-----------------------------------|
| code                           | int32  | 业务状态码，200 表示成功                    |
| message                        | string | 业务提示信息                            |
| data                           | object | VLAN 接口 IP 信息                     |
| data.interface_ip_list         | array  | VLAN 与 IP 映射列表                    |
| data.interface_ip_list.vlan_id | int32  | VLAN ID（0 代表物理接口 eth0）也就是 OOB 管理口 |
| data.interface_ip_list.ip      | string | IP 地址                             |

### 3.3 模式切换

#### 3.3.1 查询交换机二层/三层模式

##### 接口描述

查询当前交换机是二层模式还是三层模式

##### 请求地址

/v1/preset/mode/get

##### 请求方法

GET

##### 请求参数示例

```
1 {}
```

##### 请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

##### 响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {
5         "mode": "layer3"
6     }
7 }
```

##### 响应参数说明

| 参数名称      | 参数类型   | 描述                              |
|-----------|--------|---------------------------------|
| code      | int32  | 业务状态码，200 表示成功                  |
| message   | string | 业务提示信息                          |
| data.mode | string | 当前模式：layer2（二层模式）或 layer3（三层模式） |

### 3.3.2 设置交换机二层/三层模式

#### 接口描述

切换交换机的二层/三层模式

#### 请求地址

/v1/preset/mode/set

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "mode": "layer3"
3 }
```

#### 请求参数说明

| 参数名称 | 参数类型   | 是否必填 | 描述                                  |
|------|--------|------|-------------------------------------|
| mode | string | True | 目标模式: layer2 (二层模式) 或 layer3 (三层模式) |

#### 响应示例

```
1 {
2     "code": 200,
3     "message": "ok"
4 }
```

#### 响应参数说明

| 参数名称    | 参数类型   | 描述              |
|---------|--------|-----------------|
| code    | int32  | 业务状态码, 200 表示成功 |
| message | string | 业务提示信息          |

## 4 面板管理

### 4.1 输入源发现

#### 4.1.1 添加服务器发现

##### 接口描述

添加服务器发现

##### 请求地址

/v1/input/server/add

##### 请求方法

POST

##### 请求参数示例

```
1 {
2     "dis_g_name": "public",
3     "dis_s_ip": "192.168.30.125"
4 }
```

##### 请求参数说明

| 参数名称       | 参数类型   | 是否必填 | 描述          |
|------------|--------|------|-------------|
| dis_g_name | string | 是    | 发现组名称       |
| dis_s_ip   | string | 是    | 发现服务器 IP 地址 |

##### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

##### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

## 4.1.2 添加手动发现

### 接口描述

添加手动发现

### 请求地址

/v1/input/manual/add

### 请求方法

POST

### 请求参数示例

```
1 {
2     "manuals": [
3         {
4             "dis_g_name": "public",
5             "dis_d_ip": "192.168.40.23"
6         }
7     ]
8 }
```

### 请求参数说明

| 参数名称                 | 参数类型   | 是否必填 | 描述         |
|----------------------|--------|------|------------|
| manuals              | array  | 是    | 手动发现信息数组   |
| manuals[].dis_g_name | string | 是    | 发现组名称      |
| manuals[].dis_d_ip   | string | 是    | 发现设备 IP 地址 |

### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

### 4.1.3 添加自动发现

接口描述

添加自动发现

请求地址

/v1/input/auto/add

请求方法

POST

请求参数示例

```
1 {
2     "dis_g_name": "public"
3 }
```

请求参数说明

| 参数名称       | 参数类型   | 是否必填 | 描述    |
|------------|--------|------|-------|
| dis_g_name | string | 是    | 发现组名称 |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

### 4.1.4 服务器发现查询

接口描述

服务器输入发现

请求地址

/v1/input/server/discovery

请求方法

GET

请求参数示例

```
1 {
2     "dis_g_name": "group1",
3     "dis_s_ip": "192.168.1.1"
4 }
```

请求参数说明

| 参数名称       | 参数类型   | 是否必填 | 描述          |
|------------|--------|------|-------------|
| dis_g_name | string | 是    | 发现组名称       |
| dis_s_ip   | string | 是    | 发现服务器 IP 地址 |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "inputs": [
6             {
7                 "name": "input1",
8                 "alias": "输入 1",
9                 "c_name": "通道 1",
10                "d_name": "设备 1",
11                "d_ip": "192.168.1.100",
12                "d_port": 554,
13                "url": "192.168.1.100:554",
14                "enable": true,
15                "dis_method": "server",
16                "dis_g_name": "group1",
17                "dis_d_ip": "",
18                "dis_s_ip": "192.168.1.1",
19                "vlan_id": 100,
20                "switcher_id": "00000000000000"
21            }
22        ]
23    }
24 }
```

响应参数说明

| 参数名称 | 参数类型 | 描述           |
|------|------|--------------|
| code | int  | 状态码，200 表示成功 |

|                           |        |             |
|---------------------------|--------|-------------|
| message                   | string | 操作结果描述      |
| data                      | object | 返回数据        |
| data.inputs               | array  | 输入列表        |
| data.inputs[].name        | string | 输入名称        |
| data.inputs[].alias       | string | 输入别名        |
| data.inputs[].c_name      | string | 通道名称        |
| data.inputs[].d_name      | string | 设备名称        |
| data.inputs[].d_ip        | string | 设备 IP 地址    |
| data.inputs[].d_port      | int    | 设备端口        |
| data.inputs[].url         | string | 输入 URL      |
| data.inputs[].enable      | bool   | 是否启用        |
| data.inputs[].dis_method  | string | 发现方法        |
| data.inputs[].dis_g_name  | string | 发现组名称       |
| data.inputs[].dis_d_ip    | string | 发现设备 IP 地址  |
| data.inputs[].dis_s_ip    | string | 发现服务器 IP 地址 |
| data.inputs[].vlan_id     | int    | VLAN ID     |
| data.inputs[].switcher_id | string | 交换机序列号      |

### 4.1.5 手动发现查询

#### 接口描述

手动输入发现

#### 请求地址

/v1/input/manual/discovery

#### 请求方法

GET

#### 请求参数示例

无

#### 请求参数说明

无

#### 响应示例

```
1 {
```

```

2  "code": 200,
3  "message": "Success",
4  "data": {
5      "inputs": [
6          {
7              "name": "input2",
8              "alias": "输入 2",
9              "c_name": "通道 2",
10             "d_name": "设备 2",
11             "d_ip": "192.168.1.101",
12             "d_port": 554,
13             "url": "192.168.1.101:554",
14             "enable": true,
15             "dis_method": "manual",
16             "dis_g_name": "group1",
17             "dis_d_ip": "192.168.1.101",
18             "dis_s_ip": "",
19             "vlan_id": 100,
20             "switcher_id": "00000000000000"
21         }
22     ]
23 }
24 }

```

## 响应参数说明

| 参数名称                      | 参数类型   | 描述           |
|---------------------------|--------|--------------|
| code                      | int    | 状态码，200 表示成功 |
| message                   | string | 操作结果描述       |
| data                      | object | 返回数据         |
| data.inputs               | array  | 输入列表         |
| data.inputs[].name        | string | 输入名称         |
| data.inputs[].alias       | string | 输入别名         |
| data.inputs[].c_name      | string | 通道名称         |
| data.inputs[].d_name      | string | 设备名称         |
| data.inputs[].d_ip        | string | 设备 IP 地址     |
| data.inputs[].d_port      | int    | 设备端口         |
| data.inputs[].url         | string | 输入 URL       |
| data.inputs[].enable      | bool   | 是否启用         |
| data.inputs[].dis_method  | string | 发现方法         |
| data.inputs[].dis_g_name  | string | 发现组名称        |
| data.inputs[].dis_d_ip    | string | 发现设备 IP 地址   |
| data.inputs[].dis_s_ip    | string | 发现服务器 IP 地址  |
| data.inputs[].vlan_id     | int    | VLAN ID      |
| data.inputs[].switcher_id | string | 交换机序列号       |

## 4.1.6 自动发现查询

### 接口描述

自动输入发现

### 请求地址

/v1/input/auto/discovery

### 请求方法

GET

### 请求参数示例

```
1 {
2     "dis_g_name": "group1"
3 }
```

### 请求参数说明

| 参数名称       | 参数类型   | 是否必填 | 描述    |
|------------|--------|------|-------|
| dis_g_name | string | 是    | 发现组名称 |

### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "inputs": [
6             {
7                 "name": "input3",
8                 "alias": "输入 3",
9                 "c_name": "通道 3",
10                "d_name": "设备 3",
11                "d_ip": "192.168.1.102",
12                "d_port": 554,
13                "url": "192.168.1.102:554",
14                "enable": true,
15                "dis_method": "auto",
16                "dis_g_name": "group1",
17                "dis_d_ip": "192.168.1.102",
18                "dis_s_ip": "",
19                "vlan_id": 100,
20                "switcher_id": "00000000000000"
21            }
22        ]
23    }
24 }
```

响应参数说明

| 参数名称                      | 参数类型   | 描述           |
|---------------------------|--------|--------------|
| code                      | int    | 状态码，200 表示成功 |
| message                   | string | 操作结果描述       |
| data                      | object | 返回数据         |
| data.inputs               | array  | 输入列表         |
| data.inputs[].name        | string | 输入名称         |
| data.inputs[].alias       | string | 输入别名         |
| data.inputs[].c_name      | string | 通道名称         |
| data.inputs[].d_name      | string | 设备名称         |
| data.inputs[].d_ip        | string | 设备 IP 地址     |
| data.inputs[].d_port      | int    | 设备端口         |
| data.inputs[].url         | string | 输入 URL       |
| data.inputs[].enable      | bool   | 是否启用         |
| data.inputs[].dis_method  | string | 发现方法         |
| data.inputs[].dis_g_name  | string | 发现组名称        |
| data.inputs[].dis_d_ip    | string | 发现设备 IP 地址   |
| data.inputs[].dis_s_ip    | string | 发现服务器 IP 地址  |
| data.inputs[].vlan_id     | int    | VLAN ID      |
| data.inputs[].switcher_id | string | 交换机序列号       |

4.1.7 删除发现项

接口描述

删除发现的输入

请求地址

/v1/input/discovery/delete

请求方法

POST

请求参数示例

```
1 {
2   "ids": [1, 2, 3]
3 }
```

请求参数说明

| 参数名称 | 参数类型  | 是否必填 | 描述           |
|------|-------|------|--------------|
| ids  | array | 是    | 要删除的发现 ID 数组 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.1.8 获取所有发现项

接口描述

获取所有发现的输入

请求地址

/v1/input/discovery/all

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
```

```

4   "data": {
5       "manuals": [
6           {
7               "dis_id": 3,
8               "dis_g_name": "public",
9               "dis_d_ip": "192.168.40.23"
10          }
11      ],
12      "auto_servers": [
13          {
14              "dis_id": 1,
15              "dis_method": "auto",
16              "dis_g_name": "public",
17              "dis_d_ip": "",
18              "dis_s_ip": ""
19          },
20          {
21              "dis_id": 2,
22              "dis_method": "auto",
23              "dis_g_name": "public1",
24              "dis_d_ip": "",
25              "dis_s_ip": ""
26          },
27          {
28              "dis_id": 4,
29              "dis_method": "server",
30              "dis_g_name": "public",
31              "dis_d_ip": "",
32              "dis_s_ip": "192.43.56.2"
33          }
34      ]
35  }
36 }

```

## 响应参数说明

| 参数名称                           | 参数类型   | 描述           |
|--------------------------------|--------|--------------|
| code                           | int    | 状态码，200 表示成功 |
| message                        | string | 操作结果描述       |
| data                           | object | 返回数据         |
| data.manuals                   | array  | 手动发现列表       |
| data.manuals[].dis_id          | int    | 发现 ID        |
| data.manuals[].dis_g_name      | string | 发现组名称        |
| data.manuals[].dis_d_ip        | string | 发现设备 IP 地址   |
| data.auto_servers              | array  | 自动服务器发现列表    |
| data.auto_servers[].dis_id     | int    | 发现 ID        |
| data.auto_servers[].dis_method | string | 发现方法         |
| data.auto_servers[].dis_g_name | string | 发现组名称        |
| data.auto_servers[].dis_d_ip   | string | 发现设备 IP 地址   |
| data.auto_servers[].dis_s_ip   | string | 发现服务器 IP 地址  |

## 4.2 输入管理

### 4.2.1 修改输入源

#### 接口描述

修改输入源

#### 请求地址

/v1/input/modify

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "id": 1,
3     "alias": "新输入别名"
4 }
```

#### 请求参数说明

| 参数名称  | 参数类型   | 是否必填 | 描述    |
|-------|--------|------|-------|
| id    | int    | 是    | 输入 ID |
| alias | string | 否    | 输入别名  |

#### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

#### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

## 4.3 输出管理

### 4.3.1 添加输出

#### 接口描述

添加输出

#### 请求地址

/v1/output/add

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "outputs": [
3         {
4             "name": "output1",
5             "alias": "输出 1",
6             "c_name": "通道 1",
7             "g_name": "group1",
8             "router": true,
9             "dis_s_ip": "192.168.1.1",
10            "vlan_id": 100
11        }
12    ]
13 }
```

#### 请求参数说明

| 参数名称               | 参数类型   | 是否必填 | 描述          |
|--------------------|--------|------|-------------|
| outputs            | array  | 是    | 输出列表        |
| outputs[].name     | string | 是    | 输出名称        |
| outputs[].alias    | string | 是    | 输出别名        |
| outputs[].c_name   | string | 是    | 通道名称        |
| outputs[].g_name   | string | 是    | 组名称         |
| outputs[].router   | bool   | 是    | 是否路由        |
| outputs[].dis_s_ip | string | 是    | 发现服务器 IP 地址 |
| outputs[].vlan_id  | int    | 是    | VLAN ID     |

#### 响应示例

```
1 {
```

```
2  "code": 200,
3  "message": "Success",
4  "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.3.2 删除输出

接口描述

删除输出

请求地址

/v1/output/delete

请求方法

POST

请求参数示例

```
1 {
2   "ids": [1, 2, 3]
3 }
```

请求参数说明

| 参数名称 | 参数类型  | 是否必填 | 描述           |
|------|-------|------|--------------|
| ids  | array | 是    | 要删除的输出 ID 数组 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称 | 参数类型 | 描述 |
|------|------|----|
|------|------|----|

|         |        |              |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

### 4.3.3 修改输出

#### 接口描述

修改输出

#### 请求地址

/v1/output/modify

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "id": 1,
3   "name": "new_name",
4   "g_name": "new_group",
5   "router": false,
6   "vlan_id": 200
7 }
```

#### 请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述          |
|----------|--------|------|-------------|
| id       | int    | 是    | 输出 ID       |
| name     | string | 否    | 输出名称        |
| g_name   | string | 否    | 组名称         |
| dis_s_ip | string | 否    | 发现服务器 IP 地址 |
| router   | bool   | 否    | 是否路由        |
| vlan_id  | int    | 否    | VLAN ID     |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.3.4 查询输出

接口描述

查询输出

请求地址

/v1/output/query

请求方法

GET

请求参数示例

```
1 {
2   "id": 1
3 }
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| id   | int  | 是    | 输出 ID |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "id": 1,
6     "name": "output1",
7     "alias": "输出 1",
8     "c_name": "通道 1",
9     "g_name": "group1",
10    "smooth": 0,
11    "router": true,
12    "codec_k": 0,
13    "dis_s_ip": "192.168.1.1",
14    "locked": false,
15    "enable": true,
```

```
16     "vlan_id": 100,
17     "vlan_alias": "VLAN100",
18     "vlan_color": "#0000FF",
19     "switcher_id": "00000000000000",
20     "connect_num": 0,
21     "create_time": "2026-04-21 00:00:00.000000",
22     "update_time": "2026-04-21 00:00:00.000000"
23 }
24 }
```

响应参数说明

| 参数名称             | 参数类型   | 描述           |
|------------------|--------|--------------|
| code             | int    | 状态码，200 表示成功 |
| message          | string | 操作结果描述       |
| data             | object | 输出信息         |
| data.id          | int    | 输出 ID        |
| data.name        | string | 输出名称         |
| data.alias       | string | 输出别名         |
| data.c_name      | string | 通道名称         |
| data.g_name      | string | 组名称          |
| data.smooth      | int    | 平滑度          |
| data.router      | bool   | 是否路由         |
| data.codec_k     | int    | 编解码器保持连接     |
| data.dis_s_ip    | string | 发现服务器 IP 地址  |
| data.locked      | bool   | 是否锁定         |
| data.enable      | bool   | 是否启用         |
| data.vlan_id     | int    | VLAN ID      |
| data.vlan_alias  | string | VLAN 别名      |
| data.vlan_color  | string | VLAN 颜色      |
| data.switcher_id | string | 交换机序列号       |
| data.connect_num | int    | 连接数          |
| data.create_time | string | 创建时间         |
| data.update_time | string | 更新时间         |

4.3.5 获取所有输出

接口描述

查询所有输出

请求地址

/v1/output/query/all

## 请求方法

GET

## 请求参数示例

无

## 请求参数说明

无

## 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "outputs": [
6       {
7         "id": 1,
8         "name": "output1",
9         "alias": "输出 1",
10        "c_name": "通道 1",
11        "g_name": "group1",
12        "smooth": 0,
13        "router": true,
14        "codec_k": 0,
15        "dis_s_ip": "192.168.1.1",
16        "locked": false,
17        "enable": true,
18        "vlan_id": 100,
19        "vlan_alias": "VLAN100",
20        "vlan_color": "#0000FF",
21        "switcher_id": "00000000000000",
22        "connect_num": 0,
23        "create_time": "2026-04-21 00:00:00.000000",
24        "update_time": "2026-04-21 00:00:00.000000"
25      },
26      {
27        "id": 2,
28        "name": "output2",
29        "alias": "输出 2",
30        "c_name": "通道 2",
31        "g_name": "group1",
32        "smooth": 0,
33        "router": true,
34        "codec_k": 0,
35        "dis_s_ip": "192.168.1.1",
36        "locked": false,
37        "enable": true,
38        "vlan_id": 100,
39        "vlan_alias": "VLAN100",
40        "vlan_color": "#0000FF",
41        "switcher_id": "00000000000000",
42        "connect_num": 0,
```

```
43         "create_time": "2026-04-21 00:00:00.000000",
44         "update_time": "2026-04-21 00:00:00.000000"
45     }
46 ]
47 }
48 }
```

响应参数说明

| 参数名称                       | 参数类型   | 描述           |
|----------------------------|--------|--------------|
| code                       | int    | 状态码，200 表示成功 |
| message                    | string | 操作结果描述       |
| data                       | object | 返回数据         |
| data.outputs               | array  | 输出列表         |
| data.outputs[].id          | int    | 输出 ID        |
| data.outputs[].name        | string | 输出名称         |
| data.outputs[].alias       | string | 输出别名         |
| data.outputs[].c_name      | string | 通道名称         |
| data.outputs[].g_name      | string | 组名称          |
| data.outputs[].smooth      | int    | 平滑度          |
| data.outputs[].router      | bool   | 是否路由         |
| data.outputs[].codec_k     | int    | 编解码器保持连接     |
| data.outputs[].dis_s_ip    | string | 发现服务器 IP 地址  |
| data.outputs[].locked      | bool   | 是否锁定         |
| data.outputs[].enable      | bool   | 是否启用         |
| data.outputs[].vlan_id     | int    | VLAN ID      |
| data.outputs[].vlan_alias  | string | VLAN 别名      |
| data.outputs[].vlan_color  | string | VLAN 颜色      |
| data.outputs[].switcher_id | string | 交换机序列号       |
| data.outputs[].connect_num | int    | 连接数          |
| data.outputs[].create_time | string | 创建时间         |
| data.outputs[].update_time | string | 更新时间         |

4.3.6 获取默认输出

接口描述

查询默认输出

请求地址

/v1/output/query/default

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "name": "output1"
6   }
7 }
```

响应参数说明

| 参数名称      | 参数类型   | 描述           |
|-----------|--------|--------------|
| code      | int    | 状态码，200 表示成功 |
| message   | string | 操作结果描述       |
| data      | object | 默认输出信息       |
| data.name | string | 默认输出名称       |

4.4 面板基础操作

4.4.1 添加面板

接口描述

添加面板

请求地址

/v1/panel/add

请求方法

POST

请求参数示例

```
1 {
2   "name": "panel1"
3 }
```

请求参数说明

| 参数名称 | 参数类型   | 是否必填 | 描述   |
|------|--------|------|------|
| name | string | 是    | 面板名称 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.4.2 删除面板

接口描述

删除面板

请求地址

/v1/panel/delete

请求方法

POST

请求参数示例

```
1 {
2   "ids": [1, 2, 3]
3 }
```

请求参数说明

| 参数名称 | 参数类型  | 是否必填 | 描述       |
|------|-------|------|----------|
| ids  | array | 是    | 面板 ID 列表 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.4.3 修改面板

接口描述

修改面板

请求地址

/v1/panel/modify

请求方法

POST

请求参数示例

```
1 {
2   "id": 1,
3   "name": "new_panel_name"
4 }
```

请求参数说明

| 参数名称 | 参数类型   | 是否必填 | 描述    |
|------|--------|------|-------|
| id   | int    | 是    | 面板 ID |
| name | string | 是    | 面板名称  |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.4.4 展开面板

接口描述

扩展面板

请求地址

/v1/panel/expand

请求方法

POST

请求参数示例

```
1 {
2   "id": 1,
3   "expand": true
4 }
```

请求参数说明

| 参数名称   | 参数类型 | 是否必填 | 描述    |
|--------|------|------|-------|
| id     | int  | 是    | 面板 ID |
| expand | bool | 是    | 是否扩展  |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

5 }

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.4.5 查询面板

接口描述

查询面板

请求地址

/v1/panel/query

请求方法

GET

请求参数示例

```
1 {
2     "id": 1
3 }
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| id   | int  | 是    | 面板 ID |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "id": 1,
6         "name": "panel1",
7         "input_num": 0,
8         "output_num": 0,
9         "create_time": "2026-04-21 00:00:00.000000",
10        "update_time": "2026-04-21 00:00:00.000000"
11    }
12 }
```

响应参数说明

| 参数名称             | 参数类型   | 描述           |
|------------------|--------|--------------|
| code             | int    | 状态码，200 表示成功 |
| message          | string | 操作结果描述       |
| data             | object | 面板信息         |
| data.id          | int    | 面板 ID        |
| data.name        | string | 面板名称         |
| data.input_num   | int    | 面板输入数量       |
| data.output_num  | int    | 面板输出数量       |
| data.create_time | string | 创建时间         |
| data.update_time | string | 更新时间         |

4.4.6 获取所有面板

接口描述

查询所有面板

请求地址

/v1/panel/query/all

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "panels": [
6       {
7         "id": 1,
8         "name": "Default",
9         "input_num": 2,
```

```
10         "output_num": 1,
11         "create_time": "2026-04-20 21:29:57.832819",
12         "update_time": "2026-04-20 21:29:57.832820"
13     },
14     {
15         "id": 2,
16         "name": "ccc",
17         "input_num": 0,
18         "output_num": 0,
19         "create_time": "2026-04-21 13:46:41.195032",
20         "update_time": "2026-04-21 13:46:41.195032"
21     },
22     {
23         "id": 3,
24         "name": "xxx",
25         "input_num": 0,
26         "output_num": 0,
27         "create_time": "2026-04-21 13:55:07.570200",
28         "update_time": "2026-04-21 13:55:07.570200"
29     }
30 ]
31 }
32 }
```

响应参数说明

| 参数名称                      | 参数类型   | 描述           |
|---------------------------|--------|--------------|
| code                      | int    | 状态码，200 表示成功 |
| message                   | string | 操作结果描述       |
| data                      | object | 返回数据         |
| data.panels               | array  | 面板列表         |
| data.panels[].id          | int    | 面板 ID        |
| data.panels[].name        | string | 面板名称         |
| data.panels[].input_num   | int    | 面板输入数量       |
| data.panels[].output_num  | int    | 面板输出数量       |
| data.panels[].create_time | string | 创建时间         |
| data.panels[].update_time | string | 更新时间         |

4.5 面板输入管理

4.5.1 添加面板输入

接口描述

添加面板输入

请求地址

/v1/panel/input/add

请求方法

POST

请求参数示例

```
1 {
2   "id": 1,
3   "inputs": [
4     {
5       "name": "input1",
6       "alias": "输入 1",
7       "c_name": "通道 1",
8       "g_name": "group1",
9       "d_ip": "192.168.1.100",
10      "d_port": 554,
11      "url": "192.168.1.100:554",
12      "enable": true,
13      "dis_method": "manual",
14      "dis_g_name": "public",
15      "dis_d_ip": "192.168.1.100",
16      "dis_s_ip": "",
17      "vlan_id": 100,
18      "switcher_id": "00000000000000"
19    }
20  ]
21 }
```

请求参数说明

| 参数名称                | 参数类型   | 是否必填 | 描述       |
|---------------------|--------|------|----------|
| id                  | int    | 是    | 面板 ID    |
| inputs[]            | array  | 是    | 输入列表     |
| inputs[].name       | string | 是    | 名称       |
| inputs[].alias      | string | 是    | 别名       |
| inputs[].c_name     | string | 是    | 频道       |
| inputs[].g_name     | string | 是    | 组名       |
| inputs[].d_ip       | string | 是    | 设备 IP    |
| inputs[].d_port     | int    | 是    | 设备端口     |
| inputs[].url        | string | 是    | url      |
| inputs[].enable     | bool   | 是    | 是否可用     |
| inputs[].dis_method | string | 是    | 发现方式     |
| inputs[].dis_g_name | string | 是    | 发现组      |
| inputs[].dis_d_ip   | string | 是    | 发现设备 IP  |
| inputs[].dis_s_ip   | string | 是    | 发现服务器 IP |

|                      |        |   |        |
|----------------------|--------|---|--------|
| inputs[].vlan_id     | int    | 是 | vlanID |
| inputs[].switcher_id | string | 是 | 交换机序列号 |

### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

## 4.5.2 移除面板输入

### 接口描述

移除面板输入

### 请求地址

/v1/panel/input/remove

### 请求方法

POST

### 请求参数示例

```
1 {
2     "id": 1,
3     "input_ids": [1, 2, 3]
4 }
```

### 请求参数说明

| 参数名称      | 参数类型  | 是否必填 | 描述       |
|-----------|-------|------|----------|
| id        | int   | 是    | 面板 ID    |
| input_ids | array | 是    | 输入 ID 数组 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.5.3 面板输入排序

接口描述

面板输入排序

请求地址

/v1/panel/input/sort

请求方法

POST

请求参数示例

```
1 {
2   "id": 1,
3   "input_ids": [3, 1, 2]
4 }
```

请求参数说明

| 参数名称      | 参数类型  | 是否必填 | 描述           |
|-----------|-------|------|--------------|
| id        | int   | 是    | 面板 ID        |
| input_ids | array | 是    | 排序后的输入 ID 数组 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

```
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.6 面板输出管理

4.6.1 添加面板输出

接口描述

添加面板输出

请求地址

/v1/panel/output/add

请求方法

POST

请求参数示例

```
1 {
2     "id": 1,
3     "output_ids": [1, 2, 3]
4 }
```

请求参数说明

| 参数名称       | 参数类型  | 是否必填 | 描述       |
|------------|-------|------|----------|
| id         | int   | 是    | 面板 ID    |
| output_ids | array | 是    | 输出 ID 数组 |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.6.2 移除面板输出

接口描述

移除面板输出

请求地址

/v1/panel/output/remove

请求方法

POST

请求参数示例

```
1 {
2     "id": 1,
3     "output_ids": [1, 2]
4 }
```

请求参数说明

| 参数名称       | 参数类型  | 是否必填 | 描述       |
|------------|-------|------|----------|
| id         | int   | 是    | 面板 ID    |
| output_ids | array | 是    | 输出 ID 数组 |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |

|      |      |       |
|------|------|-------|
| data | null | 无数据返回 |
|------|------|-------|

### 4.6.3 面板输出排序

#### 接口描述

面板输出排序

#### 请求地址

/v1/panel/output/sort

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "id": 1,
3     "output_ids": [3, 1, 2]
4 }
```

#### 请求参数说明

| 参数名称       | 参数类型  | 是否必填 | 描述           |
|------------|-------|------|--------------|
| id         | int   | 是    | 面板 ID        |
| output_ids | array | 是    | 排序后的输出 ID 数组 |

#### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

#### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

## 4.7 面板交叉切换

### 4.7.1 获取面板详情

#### 接口描述

获取面板详情

#### 请求地址

/v1/panel/detail

#### 请求方法

GET

#### 请求参数示例

```
1 {
2     "id": 1
3 }
```

#### 请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| id   | int  | 是    | 面板 ID |

#### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "id": 1,
6         "name": "Default",
7         "expand": false,
8         "panel_inputs": [
9             {
10                 "id": 1,
11                 "name": "AVX24-4 (hhyoutput1)",
12                 "alias": "AVX24-4 (hhyoutput1)",
13                 "c_name": "AVX24-4 (hhyoutput1)",
14                 "g_name": "",
15                 "d_ip": "172.30.46.14",
16                 "d_port": 5961,
17                 "url": "172.30.46.14:5961",
18                 "enable": true,
19                 "dis_method": "auto",
20                 "dis_g_name": "public",
21                 "dis_d_ip": "",
22                 "dis_s_ip": "",
```

```

23         "vlan_id": 1,
24         "switcher_id": "000000000234D2",
25         "resolution": "0x0@0HZ",
26         "vb": "0.00Mbps",
27         "ab": "0.00Mbps",
28         "multicast": true,
29         "icon_type": "",
30         "icon_name": ""
31     }
32 ],
33     "panel_outputs": [
34     {
35         "id": 1,
36         "name": "output1",
37         "alias": "output1",
38         "c_name": "output1",
39         "g_name": "public",
40         "router": false,
41         "enable": true,
42         "dis_s_ip": "",
43         "vlan_id": 1,
44         "vlan_alias": "vlan1",
45         "vlan_color": "#1D2024",
46         "switcher_id": "000000000234D2",
47         "connect_num": 0,
48         "lock": false,
49         "resolution": "0x0@0HZ",
50         "vb": "0.00Mbps",
51         "ab": "0.00Mbps",
52         "input_id": 0,
53         "linked": false,
54         "r_id": 0,
55         "r_loop": false,
56         "r_src_id": 0,
57         "r_src_uuid": "",
58         "r_src_progress": 0,
59         "icon_type": "",
60         "icon_name": "",
61         "d_ip": "172.30.46.175"
62     }
63 ],
64     "other_inputs": [],
65     "panel_templates": [],
66     "rotationlists": []
67 }
68 }

```

## 响应参数说明

| 参数名称        | 参数类型   | 描述           |
|-------------|--------|--------------|
| code        | int    | 状态码，200 表示成功 |
| message     | string | 操作结果描述       |
| data        | object | 面板详情         |
| data.id     | int    | 面板 ID        |
| data.name   | string | 面板名称         |
| data.expand | bool   | 是否展开         |

|                                  |        |             |
|----------------------------------|--------|-------------|
| data.panel_inputs                | array  | 面板输入列表      |
| data.panel_inputs[].id           | int    | 输入 ID       |
| data.panel_inputs[].name         | string | 输入名称        |
| data.panel_inputs[].alias        | string | 输入别名        |
| data.panel_inputs[].c_name       | string | 通道名称        |
| data.panel_inputs[].g_name       | string | 组名称         |
| data.panel_inputs[].d_ip         | string | 设备 IP 地址    |
| data.panel_inputs[].d_port       | int    | 设备端口        |
| data.panel_inputs[].url          | string | 输入 URL      |
| data.panel_inputs[].enable       | bool   | 是否启用        |
| data.panel_inputs[].dis_method   | string | 发现方法        |
| data.panel_inputs[].dis_g_name   | string | 发现组名称       |
| data.panel_inputs[].dis_d_ip     | string | 发现设备 IP 地址  |
| data.panel_inputs[].dis_s_ip     | string | 发现服务器 IP 地址 |
| data.panel_inputs[].vlan_id      | int    | VLAN ID     |
| data.panel_inputs[].switcher_id  | string | 交换机序列号      |
| data.panel_inputs[].resolution   | string | 分辨率         |
| data.panel_inputs[].vb           | string | 视频比特率       |
| data.panel_inputs[].ab           | string | 音频比特率       |
| data.panel_inputs[].multicast    | bool   | 是否组播        |
| data.panel_inputs[].icon_type    | string | 图标类型        |
| data.panel_inputs[].icon_name    | string | 图标名称        |
| data.panel_outputs               | array  | 面板输出列表      |
| data.panel_outputs[].id          | int    | 输出 ID       |
| data.panel_outputs[].name        | string | 输出名称        |
| data.panel_outputs[].alias       | string | 输出别名        |
| data.panel_outputs[].c_name      | string | 通道名称        |
| data.panel_outputs[].g_name      | string | 组名称         |
| data.panel_outputs[].router      | bool   | 是否路由        |
| data.panel_outputs[].enable      | bool   | 是否启用        |
| data.panel_outputs[].dis_s_ip    | string | 发现服务器 IP 地址 |
| data.panel_outputs[].vlan_id     | int    | VLAN ID     |
| data.panel_outputs[].vlan_alias  | string | VLAN 别名     |
| data.panel_outputs[].vlan_color  | string | VLAN 颜色     |
| data.panel_outputs[].switcher_id | string | 交换机序列号      |
| data.panel_outputs[].connect_num | int    | 连接数         |
| data.panel_outputs[].lock        | bool   | 是否锁定        |
| data.panel_outputs[].resolution  | string | 分辨率         |
| data.panel_outputs[].vb          | string | 视频比特率       |
| data.panel_outputs[].ab          | string | 音频比特率       |

|                                     |        |             |
|-------------------------------------|--------|-------------|
| data.panel_outputs[].input_id       | int    | 输入 ID       |
| data.panel_outputs[].linked         | bool   | 是否链接        |
| data.panel_outputs[].r_id           | int    | 路由 ID       |
| data.panel_outputs[].r_loop         | bool   | 是否路由循环      |
| data.panel_outputs[].r_src_id       | int    | 路由源 ID      |
| data.panel_outputs[].r_src_uuid     | string | 路由源 UUID    |
| data.panel_outputs[].r_src_progress | int    | 路由源进度       |
| data.panel_outputs[].icon_type      | string | 图标类型        |
| data.panel_outputs[].icon_name      | string | 图标名称        |
| data.panel_outputs[].d_ip           | string | 设备 IP 地址    |
| data.other_inputs                   | array  | 其他输入列表      |
| data.other_inputs[].id              | int    | 输入 ID       |
| data.other_inputs[].name            | string | 输入名称        |
| data.other_inputs[].alias           | string | 输入别名        |
| data.other_inputs[].c_name          | string | 通道名称        |
| data.other_inputs[].g_name          | string | 组名称         |
| data.other_inputs[].d_ip            | string | 设备 IP 地址    |
| data.other_inputs[].d_port          | int    | 设备端口        |
| data.other_inputs[].url             | string | 输入 URL      |
| data.other_inputs[].enable          | bool   | 是否启用        |
| data.other_inputs[].dis_method      | string | 发现方法        |
| data.other_inputs[].dis_g_name      | string | 发现组名称       |
| data.other_inputs[].dis_d_ip        | string | 发现设备 IP 地址  |
| data.other_inputs[].dis_s_ip        | string | 发现服务器 IP 地址 |
| data.other_inputs[].vlan_id         | int    | VLAN ID     |
| data.other_inputs[].switcher_id     | string | 交换机序列号      |
| data.other_inputs[].resolution      | string | 分辨率         |
| data.other_inputs[].vb              | string | 视频比特率       |
| data.other_inputs[].ab              | string | 音频比特率       |
| data.other_inputs[].multicast       | bool   | 是否组播        |
| data.other_inputs[].icon_type       | string | 图标类型        |
| data.other_inputs[].icon_name       | string | 图标名称        |
| data.panel_templates                | array  | 面板模板列表      |
| data.rotationlists                  | array  | 轮播列表        |

## 4.7.2 输出交叉输入

### 接口描述

面板输出输入交叉

请求地址

/v1/panel/cross/output/input

请求方法

POST

请求参数示例

```
1 {
2   "id": 1,
3   "output_id": 1,
4   "input_id": 2
5 }
```

请求参数说明

| 参数名称      | 参数类型 | 是否必填 | 描述    |
|-----------|------|------|-------|
| id        | int  | 是    | 面板 ID |
| output_id | int  | 是    | 输出 ID |
| input_id  | int  | 是    | 输入 ID |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.7.3 输入选择

接口描述

面板输入选择

请求地址

/v1/panel/cross/input/select

请求方法

POST

请求参数示例

```
1 {
2     "id": 1,
3     "input_id": 1,
4     "select": true
5 }
```

请求参数说明

| 参数名称     | 参数类型 | 是否必填 | 描述    |
|----------|------|------|-------|
| id       | int  | 是    | 面板 ID |
| input_id | int  | 是    | 输入 ID |
| select   | bool | 是    | 是否全选  |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.8 面板配置

4.8.1 获取所有输入配置

接口描述

获取所有输入配置

请求地址

/v1/panel/cfg/input/all

请求方法

GET

请求参数示例

```
1 {
2     "id": 1
3 }
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| id   | int  | 是    | 面板 ID |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "cfg_inputs": [
6             {
7                 "id": 1,
8                 "name": "input1",
9                 "alias": "输入 1",
10                "icon_id": 1,
11                "front_content": "input1"
12            }
13        ]
14    }
15 }
```

响应参数说明

| 参数名称                            | 参数类型   | 描述           |
|---------------------------------|--------|--------------|
| code                            | int    | 状态码，200 表示成功 |
| message                         | string | 操作结果描述       |
| data                            | object | 返回数据         |
| data.cfg_inputs                 | array  | 输入配置列表       |
| data.cfg_inputs[].id            | int    | 输入 ID        |
| data.cfg_inputs[].name          | string | 输入名称         |
| data.cfg_inputs[].alias         | string | 输入别名         |
| data.cfg_inputs[].icon_id       | int    | 图标 ID        |
| data.cfg_inputs[].front_content | string | 前端内容         |

## 4.8.2 获取所有输出配置

### 接口描述

获取所有输出配置

### 请求地址

/v1/panel/cfg/output/all

### 请求方法

GET

### 请求参数示例

```
1 {
2     "id": 1
3 }
```

### 请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| id   | int  | 是    | 面板 ID |

### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "outputs": [
6             {
7                 "id": 1,
8                 "name": "output1",
9                 "alias": "输出 1",
10                "icon_id": 1,
11                "front_content": "output1"
12            }
13        ]
14    }
15 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | object | 返回数据         |

|                                  |        |        |
|----------------------------------|--------|--------|
| data.cfg_outputs                 | array  | 输出配置列表 |
| data.cfg_outputs[].id            | int    | 输出 ID  |
| data.cfg_outputs[].name          | string | 输出名称   |
| data.cfg_outputs[].alias         | string | 输出别名   |
| data.cfg_outputs[].icon_id       | int    | 图标 ID  |
| data.cfg_outputs[].front_content | string | 前端内容   |

### 4.8.3 应用输入配置

#### 接口描述

应用输入配置

#### 请求地址

/v1/panel/cfg/input/apply

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "id": 1,
3   "input_id": 1,
4   "icon_id": 1,
5   "front_content": "input1"
6 }
```

#### 请求参数说明

| 参数名称          | 参数类型   | 是否必填 | 描述      |
|---------------|--------|------|---------|
| id            | int    | 是    | 面板 ID   |
| input_id      | int    | 是    | 面板输入 ID |
| icon_id       | int    | 是    | 图标 ID   |
| front_content | string | 是    | 前端内容    |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.8.4 应用输出配置

接口描述

应用输出配置

请求地址

/v1/panel/cfg/output/apply

请求方法

POST

请求参数示例

```
1 {
2   "id": 1,
3   "output_id": 1,
4   "icon_id": 1,
5   "front_content": "output1"
6 }
```

请求参数说明

| 参数名称          | 参数类型   | 是否必填 | 描述      |
|---------------|--------|------|---------|
| id            | int    | 是    | 面板 ID   |
| output_id     | int    | 是    | 面板输出 ID |
| icon_id       | int    | 是    | 图标 ID   |
| front_content | string | 是    | 前端内容    |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

4.8.5 设置物理面板启用状态

接口描述

设置启用状态

请求地址

/v1/panel/cfg/enable/set

请求方法

POST

请求参数示例

```
1 {
2     "id": 1,
3     "enable": true
4 }
```

请求参数说明

| 参数名称   | 参数类型 | 是否必填 | 描述    |
|--------|------|------|-------|
| id     | int  | 是    | 面板 ID |
| enable | bool | 是    | 启用状态  |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |

|      |      |       |
|------|------|-------|
| data | null | 无数据返回 |
|------|------|-------|

## 4.8.6 查询物理面板启用状态

### 接口描述

查询启用状态

### 请求地址

/v1/panel/cfg/enable/query

### 请求方法

GET

### 请求参数示例

```
1 {
2     "id": 1
3 }
```

### 请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| id   | int  | 是    | 面板 ID |

### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "enable": true
6     }
7 }
```

### 响应参数说明

| 参数名称        | 参数类型   | 描述           |
|-------------|--------|--------------|
| code        | int    | 状态码，200 表示成功 |
| message     | string | 操作结果描述       |
| data        | object | 启用状态信息       |
| data.enable | bool   | 启用状态         |

# 5 KLS 面板管理

## 5.1 KLS 输入源发现管理

### 5.1.1 KLS 添加服务器发现

#### 接口描述

添加 KLS 服务器输入

#### 请求地址

/v1/kls/input/server/add

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "serial_number": "KLSSERIAL001",
3   "dis_g_name": "public",
4   "dis_s_ip": "192.168.1.1"
5 }
```

#### 请求参数说明

| 参数名称          | 参数类型   | 是否必填 | 描述          |
|---------------|--------|------|-------------|
| serial_number | string | 是    | 序列号         |
| dis_g_name    | string | 是    | 发现组名称       |
| dis_s_ip      | string | 是    | 发现服务器 IP 地址 |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

#### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |

|      |      |       |
|------|------|-------|
| data | null | 无数据返回 |
|------|------|-------|

### 5.1.2 KLS 添加手动发现

#### 接口描述

手动添加 KLS 输入

#### 请求地址

/v1/kls/input/manual/add

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "serial_number": "KLSSERIAL001",
3   "manuals": [
4     {
5       "dis_g_name": "public",
6       "dis_d_ip": "192.168.1.100"
7     }
8   ]
9 }
```

#### 请求参数说明

| 参数名称                 | 参数类型   | 是否必填 | 描述         |
|----------------------|--------|------|------------|
| serial_number        | string | 是    | 序列号        |
| manuals              | array  | 是    | 手动发现信息数组   |
| manuals[].dis_g_name | string | 是    | 发现组名称      |
| manuals[].dis_d_ip   | string | 是    | 发现设备 IP 地址 |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

#### 响应参数说明

| 参数名称 | 参数类型 | 描述           |
|------|------|--------------|
| code | int  | 状态码，200 表示成功 |

|         |        |        |
|---------|--------|--------|
| message | string | 操作结果描述 |
| data    | null   | 无数据返回  |

### 5.1.3 KLS 添加自动发现

#### 接口描述

自动添加 KLS 输入

#### 请求地址

/v1/kls/input/auto/add

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "serial_number": "KLSSERIAL001",
3     "dis_g_name": "public"
4 }
```

#### 请求参数说明

| 参数名称          | 参数类型   | 是否必填 | 描述    |
|---------------|--------|------|-------|
| serial_number | string | 是    | 序列号   |
| dis_g_name    | string | 是    | 发现组名称 |

#### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

#### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

## 5.1.4 KLS 删除发现项

### 接口描述

删除发现的 KLS 输入

### 请求地址

/v1/kls/input/discovery/delete

### 请求方法

POST

### 请求参数示例

```
1 {
2   "serial_number": "KLSSERIAL001",
3   "discoverys": [
4     {
5       "dis_method": "manual",
6       "dis_g_name": "public",
7       "dis_d_ip": "192.168.1.100",
8       "dis_s_ip": ""
9     }
10  ]
11 }
```

### 请求参数说明

| 参数名称                    | 参数类型   | 是否必填 | 描述          |
|-------------------------|--------|------|-------------|
| serial_number           | string | 是    | 序列号         |
| discoverys              | array  | 是    | 要删除的发现数据数组  |
| discoverys[].dis_method | string | 是    | 发现方法        |
| discoverys[].dis_g_name | string | 是    | 发现组名称       |
| discoverys[].dis_d_ip   | string | 是    | 发现设备 IP 地址  |
| discoverys[].dis_s_ip   | string | 是    | 发现服务器 IP 地址 |

### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

## 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

## 5.2 KLS 输出管理

### 5.2.1 KLS 添加输出

#### 接口描述

添加 KLS 输出

#### 请求地址

/v1/cls/output/add

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "outputs": [
3         {
4             "uuid": "485441684654546484654",
5             "name": "cls-output-1",
6             "alias": "KLS 输出 1",
7             "c_name": "通道 1",
8             "g_name": "group1",
9             "router": true,
10            "dis_s_ip": "192.168.1.1",
11            "vlan_id": 100,
12            "switcher_id": "00000000000000"
13        }
14    ]
15 }
```

#### 请求参数说明

| 参数名称           | 参数类型   | 是否必填 | 描述      |
|----------------|--------|------|---------|
| outputs        | array  | 是    | 输出列表    |
| outputs[].uuid | string | 是    | 输出 UUID |
| outputs[].name | string | 是    | 输出名称    |

|                       |        |   |             |
|-----------------------|--------|---|-------------|
| outputs[].alias       | string | 是 | 输出别名        |
| outputs[].c_name      | string | 是 | 通道名称        |
| outputs[].g_name      | string | 是 | 组名称         |
| outputs[].router      | bool   | 是 | 是否路由        |
| outputs[].dis_s_ip    | string | 是 | 发现服务器 IP 地址 |
| outputs[].vlan_id     | int    | 是 | VLAN ID     |
| outputs[].switcher_id | string | 是 | 交换机序列号      |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "outputs": [
6             {
7                 "uuid": "485441684654546484654",
8                 "name": "kls-output-1",
9                 "alias": "KLS 输出 1",
10                "c_name": "通道 1",
11                "g_name": "group1",
12                "router": true,
13                "dis_s_ip": "192.168.1.1",
14                "vlan_id": 100,
15                "switcher_id": "00000000000000"
16            }
17        ],
18        "matrixs": []
19    }
20 }
```

响应参数说明

| 参数名称                       | 参数类型   | 描述           |
|----------------------------|--------|--------------|
| code                       | int    | 状态码，200 表示成功 |
| message                    | string | 操作结果描述       |
| data                       | object | 返回数据         |
| data.outputs               | array  | 添加的输出列表      |
| data.outputs[].uuid        | string | 输出 UUID      |
| data.outputs[].name        | string | 输出名称         |
| data.outputs[].alias       | string | 输出别名         |
| data.outputs[].c_name      | string | 通道名称         |
| data.outputs[].g_name      | string | 组名称          |
| data.outputs[].router      | bool   | 是否路由         |
| data.outputs[].dis_s_ip    | string | 发现服务器 IP 地址  |
| data.outputs[].vlan_id     | int    | VLAN ID      |
| data.outputs[].switcher_id | string | 交换机序列号       |
| data.matrixs               | array  | 矩阵信息列表       |

|                           |        |           |
|---------------------------|--------|-----------|
| data.matrixs.output_uuid  | string | 输出 UUID   |
| data.matrixs[].input_name | string | 输入名称      |
| data.matrixs[].rl_uuid    | string | 轮播单 UUID  |
| data.matrixs[].rl_loop    | bool   | 轮播单是否循环播放 |

## 5.2.2 KLS 删除输出

### 接口描述

删除 KLS 输出

### 请求地址

/v1/cls/output/delete

### 请求方法

POST

### 请求参数示例

```
1 {
2   "uuids": ["485441684654546484654"]
3 }
```

### 请求参数说明

| 参数名称  | 参数类型  | 是否必填 | 描述             |
|-------|-------|------|----------------|
| uuids | array | 是    | 要删除的输出 UUID 数组 |

### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

### 5.2.3 KLS 修改输出

#### 接口描述

修改 KLS 输出

#### 请求地址

/v1/kls/output/modify

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "uuid": "485441684654546484654",
3   "name": "new-name",
4   "vlan_id": 200,
5   "g_name": "new-group",
6   "dis_s_ip": "192.168.1.2"
7 }
```

#### 请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述          |
|----------|--------|------|-------------|
| uuid     | string | 是    | 输出 UUID     |
| name     | string | 是    | 输出名称        |
| vlan_id  | int    | 是    | VLAN ID     |
| g_name   | string | 是    | 组名称         |
| dis_s_ip | string | 是    | 发现服务器 IP 地址 |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

#### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

## 5.2.4 KLS 查询输出

### 接口描述

KLS 查询输出

### 请求地址

/v1/kls/output/query

### 请求方法

GET

### 请求参数示例

```
1 {
2     "uuid": "485441684654546484654"
3 }
```

### 请求参数说明

| 参数名称 | 参数类型   | 是否必填 | 描述      |
|------|--------|------|---------|
| uuid | string | 是    | 输出 UUID |

### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "uuid": "485441684654546484654",
6         "name": "kls-output-1",
7         "alias": "KLS 输出 1",
8         "c_name": "通道 1",
9         "g_name": "group1",
10        "router": true,
11        "dis_s_ip": "192.168.1.1",
12        "vlan_id": 100,
13        "switcher_id": "00000000000000"
14    }
15 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | object | 输出信息         |

|                  |        |             |
|------------------|--------|-------------|
| data.uuid        | string | 输出 UUID     |
| data.name        | string | 输出名称        |
| data.alias       | string | 输出别名        |
| data.c_name      | string | 通道名称        |
| data.g_name      | string | 组名称         |
| data.router      | bool   | 是否路由        |
| data.dis_s_ip    | string | 发现服务器 IP 地址 |
| data.vlan_id     | int    | VLAN ID     |
| data.switcher_id | string | 交换机序列号      |

## 5.2.5 KLS 获取所有输出

### 接口描述

KLS 获取所有输出

### 请求地址

/v1/kls/output/query/all

### 请求方法

GET

### 请求参数示例

无

### 请求参数说明

无

### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "outputs": [
6       {
7         "uuid": "485441684654546484654",
8         "name": "kls-output-1",
9         "alias": "KLS 输出 1",
10        "c_name": "通道 1",
11        "g_name": "group1",
12        "router": true,
13        "dis_s_ip": "192.168.1.1",
```

```
14         "vlan_id": 100,
15         "switcher_id": "00000000000000"
16     }
17 ]
18 }
19 }
```

响应参数说明

| 参数名称                       | 参数类型   | 描述           |
|----------------------------|--------|--------------|
| code                       | int    | 状态码，200 表示成功 |
| message                    | string | 操作结果描述       |
| data                       | object | 返回数据         |
| data.outputs               | array  | 输出列表         |
| data.outputs[].uuid        | string | 输出 UUID      |
| data.outputs[].name        | string | 输出名称         |
| data.outputs[].alias       | string | 输出别名         |
| data.outputs[].c_name      | string | 通道名称         |
| data.outputs[].g_name      | string | 组名称          |
| data.outputs[].router      | bool   | 是否路由         |
| data.outputs[].dis_s_ip    | string | 发现服务器 IP 地址  |
| data.outputs[].vlan_id     | int    | VLAN ID      |
| data.outputs[].switcher_id | string | 交换机序列号       |

5.3 KLS 面板基础操作

5.3.1 KLS 添加面板

接口描述

添加 KLS 面板

请求地址

/v1/kls/panel/add

请求方法

POST

请求参数示例

```
1 {
2     "name": "kls-panel-1",
3     "uuid": "panel-uuid-1",
```

```
4   "serial_number": "KLSSERIAL001"
5 }
```

请求参数说明

| 参数名称          | 参数类型   | 是否必填 | 描述      |
|---------------|--------|------|---------|
| name          | string | 是    | 面板名称    |
| uuid          | string | 是    | 面板 UUID |
| serial_number | string | 是    | 序列号     |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

5.3.2 KLS 删除面板

接口描述

删除 KLS 面板

请求地址

/v1/kls/panel/delete

请求方法

POST

请求参数示例

```
1 {
2   "uuid": "panel-uuid-1",
3   "serial_number": "KLSSERIAL001"
4 }
```

请求参数说明

| 参数名称          | 参数类型   | 是否必填 | 描述      |
|---------------|--------|------|---------|
| uuid          | string | 是    | 面板 UUID |
| serial_number | string | 是    | 序列号     |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

5.3.3 KLS 交换机已启用

接口描述

启用 KLS 交换机

请求地址

/v1/kls/panel/switcher/enable

请求方法

POST

请求参数示例

```
1 {
2   "serial_number": "KLSSERIAL001",
3   "discoverys": [
4     {
5       "dis_method": "auto",
6       "dis_g_name": "public",
7       "dis_d_ip": "192.168.1.100",
8       "dis_s_ip": "192.168.1.1"
9     }
10  ],
11   "inputs": [
12     {
```

```

13     "name": "input1",
14     "alias": "输入 1",
15     "channel_name": "通道 1",
16     "group_name": "group1",
17     "dev_ip": "192.168.1.100",
18     "dev_port": 5961,
19     "url": "192.168.1.100:5961",
20     "enable": true,
21     "dis_method": "auto",
22     "dis_g_name": "public",
23     "dis_d_ip": "192.168.1.100",
24     "dis_s_ip": "192.168.1.1",
25     "vlan_id": 1,
26     "switcher_id": "KLSSERIAL001"
27 }
28 ],
29 "outputs": [
30     {
31         "uuid": "output-uuid-1",
32         "name": "output1",
33         "alias": "输出 1",
34         "channel_name": "通道 1",
35         "group_name": "group1",
36         "router": true,
37         "codec_keepconn": 0,
38         "dis_s_ip": "192.168.1.1",
39         "locked": false,
40         "enable": true,
41         "vlan_id": 1,
42         "switcher_id": "KLSSERIAL001",
43         "multi_add": "239.0.0.1",
44         "multi_mask": "255.255.255.0",
45         "ttl_time": 16
46     }
47 ],
48 "panels": [
49     {
50         "uuid": "panel-uuid-1",
51         "name": "Default",
52         "Icon": ""
53     }
54 ],
55 "panel_inputs": [
56     {
57         "panel_uuid": "panel-uuid-1",
58         "input": {
59             "name": "input1",
60             "alias": "输入 1",
61             "channel_name": "通道 1",
62             "group_name": "group1",
63             "dev_ip": "192.168.1.100",
64             "dev_port": 5961,
65             "url": "192.168.1.100:5961",
66             "enable": true,
67             "dis_method": "auto",
68             "dis_g_name": "public",
69             "dis_d_ip": "192.168.1.100",
70             "dis_s_ip": "192.168.1.1",
71             "vlan_id": 1,
72             "switcher_id": "KLSSERIAL001"

```

```

73     }
74   }
75 ],
76 "panel_outputs": [
77   {
78     "panel_uuid": "panel-uuid-1",
79     "output": {
80       "uuid": "output-uuid-1",
81       "name": "output1",
82       "alias": "输出 1",
83       "channel_name": "通道 1",
84       "group_name": "group1",
85       "router": true,
86       "codec_keepconn": 0,
87       "dis_s_ip": "192.168.1.1",
88       "locked": false,
89       "enable": true,
90       "vlan_id": 1,
91       "switcher_id": "KLSSERIAL001",
92       "multi_add": "239.0.0.1",
93       "multi_mask": "255.255.255.0",
94       "ttl_time": 16
95     }
96   }
97 ],
98 "panel_matrixs": [
99   {
100     "panel_uuid": "panel-uuid-1",
101     "output_uuid": "output-uuid-1",
102     "input_name": "input1",
103     "rl_uuid": "",
104     "rl_loop": false
105   }
106 ]
107 }

```

## 请求参数说明

| 参数名称                    | 参数类型   | 是否必填 | 描述       |
|-------------------------|--------|------|----------|
| serial_number           | string | 是    | 序列号      |
| discoverys              | array  | 是    | 发现列表     |
| discoverys[].dis_method | string | 是    | 发现方法     |
| discoverys[].dis_g_name | string | 是    | 发现组名称    |
| discoverys[].dis_d_ip   | string | 是    | 发现 IP    |
| discoverys[].dis_s_ip   | string | 是    | 发现交换机 IP |
| inputs                  | array  | 是    | 输入列表     |
| inputs[].name           | string | 是    | 输入名称     |
| inputs[].alias          | string | 是    | 输入别名     |
| inputs[].channel_name   | string | 是    | 输入通道名称   |
| inputs[].group_name     | string | 是    | 输入组名称    |
| inputs[].dev_ip         | string | 是    | 输入 IP    |
| inputs[].dev_port       | int    | 是    | 输入端口     |

|                             |        |   |            |
|-----------------------------|--------|---|------------|
| inputs[].url                | string | 是 | 输入 URL     |
| inputs[].enable             | bool   | 是 | 输入启用状态     |
| inputs[].dis_method         | string | 是 | 输入发现方法     |
| inputs[].dis_g_name         | string | 是 | 输入发现组名称    |
| inputs[].dis_d_ip           | string | 是 | 输入发现 IP    |
| inputs[].dis_s_ip           | string | 是 | 输入发现交换机 IP |
| inputs[].vlan_id            | int    | 是 | VLAN ID    |
| inputs[].switcher_id        | string | 是 | 序列号        |
| outputs                     | array  | 是 | 输出列表       |
| outputs[].uuid              | string | 是 | 输出 UUID    |
| outputs[].name              | string | 是 | 输出名称       |
| outputs[].alias             | string | 是 | 输出别名       |
| outputs[].channel_name      | string | 是 | 输出通道名称     |
| outputs[].group_name        | string | 是 | 输出组名称      |
| outputs[].router            | bool   | 是 | 输出路由       |
| outputs[].codec_keepconn    | int    | 是 | 输出保持连接     |
| outputs[].dis_s_ip          | string | 是 | 输出发现交换机 IP |
| outputs[].locked            | bool   | 是 | 输出锁定       |
| outputs[].enable            | bool   | 是 | 输出启用状态     |
| outputs[].vlan_id           | int    | 是 | VLAN ID    |
| outputs[].switcher_id       | string | 是 | 序列号        |
| outputs[].multi_add         | string | 是 | 组播地址       |
| outputs[].multi_mask        | string | 是 | 组播掩码       |
| outputs[].ttl_time          | int    | 是 | TTL 时间     |
| panels                      | array  | 是 | 面板列表       |
| panels[].uuid               | string | 是 | 面板 UUID    |
| panels[].name               | string | 是 | 面板名称       |
| panels[].lcon               | string | 是 | 面板图标       |
| panel_inputs                | array  | 是 | 面板输入列表     |
| panel_inputs[].panel_uuid   | string | 是 | 面板 UUID    |
| panel_inputs[].input        | object | 是 | 输入对象       |
| panel_outputs               | array  | 是 | 面板输出列表     |
| panel_outputs[].panel_uuid  | string | 是 | 面板 UUID    |
| panel_outputs[].output      | object | 是 | 输出对象       |
| panel_matrixs               | array  | 是 | 面板矩阵列表     |
| panel_matrixs[].panel_uuid  | string | 是 | 面板 UUID    |
| panel_matrixs[].output_uuid | string | 是 | 输出 UUID    |
| panel_matrixs[].input_name  | string | 是 | 输入名称       |
| panel_matrixs[].rl_uuid     | string | 是 | 轮播单 UUID   |
| panel_matrixs[].rl_loop     | bool   | 是 | 轮播单循环      |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

5.3.4 KLS 删除设备

接口描述

删除 KLS 设备

请求地址

/v1/kls/panel/device/delete

请求方法

POST

请求参数示例

```
1 {
2   "serial_number": "KLSSERIAL001"
3 }
```

请求参数说明

| 参数名称          | 参数类型   | 是否必填 | 描述  |
|---------------|--------|------|-----|
| serial_number | string | 是    | 序列号 |

响应示例

```
1 {
2   "code": 200,
3   "message": "删除 KLS 设备成功",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

5.4 KLS 面板输入管理

5.4.1 KLS 添加面板输入

接口描述

添加 KLS 面板输入

请求地址

/v1/kls/panel/input/add

请求方法

POST

请求参数示例

```
1 {
2     "panel_uuid": "panel-uuid-1",
3     "inputs": [
4         {
5             "name": "input1",
6             "alias": "输入 1",
7             "c_name": "通道 1",
8             "g_name": "group1",
9             "d_ip": "192.168.1.100",
10            "d_port": 5961,
11            "url": "192.168.1.100:5961",
12            "enable": true,
13            "dis_method": "auto",
14            "dis_g_name": "public",
15            "dis_d_ip": "192.168.1.100",
16            "dis_s_ip": "192.168.1.1",
17            "vlan_id": 1,
18            "switcher_id": "KLSSERIAL001"
19        }
20    ]
21 }
```

请求参数说明

| 参数名称                 | 参数类型   | 是否必填 | 描述         |
|----------------------|--------|------|------------|
| panel_uuid           | string | 是    | 面板 UUID    |
| inputs               | array  | 是    | 输入对象数组     |
| inputs[].name        | string | 是    | 输入名称       |
| inputs[].alias       | string | 是    | 输入别名       |
| inputs[].c_name      | string | 是    | 输入通道名称     |
| inputs[].g_name      | string | 是    | 输入组名称      |
| inputs[].d_ip        | string | 是    | 输入 IP      |
| inputs[].d_port      | int    | 是    | 输入端口       |
| inputs[].url         | string | 是    | 输入 URL     |
| inputs[].enable      | bool   | 是    | 输入启用状态     |
| inputs[].dis_method  | string | 是    | 输入发现方法     |
| inputs[].dis_g_name  | string | 是    | 输入发现组名称    |
| inputs[].dis_d_ip    | string | 是    | 输入发现 IP    |
| inputs[].dis_s_ip    | string | 是    | 输入发现交换机 IP |
| inputs[].vlan_id     | int    | 是    | 输入 VLAN ID |
| inputs[].switcher_id | string | 是    | 序列号        |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

5.4.2 KLS 移除面板输入

接口描述

移除 KLS 面板输入

请求地址

/v1/kls/panel/input/remove

请求方法

POST

请求参数示例

```
1 {
2   "panel_uuid": "panel-uuid-1",
3   "input_names": ["input1", "input2"]
4 }
```

请求参数说明

| 参数名称        | 参数类型   | 是否必填 | 描述      |
|-------------|--------|------|---------|
| panel_uuid  | string | 是    | 面板 UUID |
| input_names | array  | 是    | 输入名称数组  |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

5.5 KLS 面板输出管理

5.5.1 KLS 添加面板输出

接口描述

添加 KLS 面板输出

请求地址

/v1/kls/panel/output/add

请求方法

POST

请求参数示例

```
1 {
2   "panel_uuid": "panel-uuid-1",
3   "output_uuids": ["output-uuid-1", "output-uuid-2"]
4 }
```

请求参数说明

| 参数名称         | 参数类型   | 是否必填 | 描述         |
|--------------|--------|------|------------|
| panel_uuid   | string | 是    | 面板 UUID    |
| output_uuids | array  | 是    | 输出 UUID 数组 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

5.5.2 KLS 移除面板输出

接口描述

移除 KLS 面板输出

请求地址

/v1/cls/panel/output/remove

请求方法

POST

请求参数示例

```
1 {
2   "panel_uuid": "panel-uuid-1",
3   "output_uuids": ["output-uuid-1"]
4 }
```

请求参数说明

| 参数名称         | 参数类型   | 是否必填 | 描述         |
|--------------|--------|------|------------|
| panel_uuid   | string | 是    | 面板 UUID    |
| output_uuids | array  | 是    | 输出 UUID 数组 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

5.6 KLS 面板交叉切换

5.6.1 KLS 获取面板详情

接口描述

获取 KLS 面板详情

请求地址

/v1/kls/panel/detail

请求方法

GET

请求参数示例

```
1 {
```

```
2   "panel_uuid": "panel-uuid-1"
3 }
```

请求参数说明

| 参数名称       | 参数类型   | 是否必填 | 描述      |
|------------|--------|------|---------|
| panel_uuid | string | 是    | 面板 UUID |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "uuid": "panel-uuid-1",
6     "panel_inputs": [
7       {
8         "name": "input1",
9         "alias": "输入 1",
10        "c_name": "通道 1",
11        "g_name": "group1",
12        "d_ip": "192.168.1.100",
13        "d_port": 5961,
14        "url": "192.168.1.100:5961",
15        "enable": true,
16        "dis_method": "auto",
17        "dis_g_name": "public",
18        "dis_d_ip": "192.168.1.100",
19        "dis_s_ip": "192.168.1.1",
20        "vlan_id": 1,
21        "switcher_id": "KLSSERIAL001",
22        "resolution": "1920x1080@60HZ",
23        "vb": "50.00Mbps",
24        "ab": "1.50Mbps",
25        "multicast": true
26      }
27    ],
28    "panel_outputs": [
29      {
30        "uuid": "output-uuid-1",
31        "name": "output1",
32        "alias": "输出 1",
33        "c_name": "通道 1",
34        "g_name": "group1",
35        "router": true,
36        "enable": true,
37        "dis_s_ip": "192.168.1.1",
38        "vlan_id": 1,
39        "vlan_alias": "VLAN1",
40        "vlan_color": "#0000FF",
41        "switcher_id": "KLSSERIAL001",
42        "connect_num": 0,
43        "lock": false,
44        "resolution": "1920x1080@60HZ",
45        "vb": "50.00Mbps",
46        "ab": "1.50Mbps",
47        "input_name": "input1",
48        "linked": false,
```

```

49         "r_uuid": "",
50         "r_loop": false,
51         "r_src_id": "",
52         "r_src_uuid": "",
53         "r_src_progress": 0,
54         "d_ip": "192.168.1.200"
55     }
56 ],
57     "other_inputs": [],
58     "panel_templates": [],
59     "rotationlists": []
60 }
61 }

```

## 响应参数说明

| 参数名称                            | 参数类型   | 描述           |
|---------------------------------|--------|--------------|
| code                            | int    | 状态码，200 表示成功 |
| message                         | string | 操作结果描述       |
| data                            | object | 面板详情         |
| data.uuid                       | string | 面板 UUID      |
| data.panel_inputs               | array  | 面板输入列表       |
| data.panel_inputs[].name        | string | 输入名称         |
| data.panel_inputs[].alias       | string | 输入别名         |
| data.panel_inputs[].c_name      | string | 通道名称         |
| data.panel_inputs[].g_name      | string | 组名称          |
| data.panel_inputs[].d_ip        | string | 设备 IP 地址     |
| data.panel_inputs[].d_port      | int    | 设备端口         |
| data.panel_inputs[].url         | string | 输入 URL       |
| data.panel_inputs[].enable      | bool   | 是否启用         |
| data.panel_inputs[].dis_method  | string | 发现方法         |
| data.panel_inputs[].dis_g_name  | string | 发现组名称        |
| data.panel_inputs[].dis_d_ip    | string | 发现设备 IP 地址   |
| data.panel_inputs[].dis_s_ip    | string | 发现服务器 IP 地址  |
| data.panel_inputs[].vlan_id     | int    | VLAN ID      |
| data.panel_inputs[].switcher_id | string | 序列号          |
| data.panel_inputs[].resolution  | string | 分辨率          |
| data.panel_inputs[].vb          | string | 视频比特率        |
| data.panel_inputs[].ab          | string | 音频比特率        |
| data.panel_inputs[].multicast   | bool   | 是否组播         |
| data.panel_outputs              | array  | 面板输出列表       |
| data.panel_outputs[].uuid       | string | 输出 UUID      |
| data.panel_outputs[].name       | string | 输出名称         |
| data.panel_outputs[].alias      | string | 输出别名         |
| data.panel_outputs[].c_name     | string | 通道名称         |

|                                     |        |             |
|-------------------------------------|--------|-------------|
| data.panel_outputs[].g_name         | string | 组名称         |
| data.panel_outputs[].router         | bool   | 是否路由        |
| data.panel_outputs[].enable         | bool   | 是否启用        |
| data.panel_outputs[].dis_s_ip       | string | 发现服务器 IP 地址 |
| data.panel_outputs[].vlan_id        | int    | VLAN ID     |
| data.panel_outputs[].vlan_alias     | string | VLAN 别名     |
| data.panel_outputs[].vlan_color     | string | VLAN 颜色     |
| data.panel_outputs[].switcher_id    | string | 序列号         |
| data.panel_outputs[].connect_num    | int    | 连接数         |
| data.panel_outputs[].lock           | bool   | 是否锁定        |
| data.panel_outputs[].resolution     | string | 分辨率         |
| data.panel_outputs[].vb             | string | 视频比特率       |
| data.panel_outputs[].ab             | string | 音频比特率       |
| data.panel_outputs[].input_name     | string | 输入名称        |
| data.panel_outputs[].linked         | bool   | 是否链接        |
| data.panel_outputs[].r_uuid         | string | 轮播单 UUID    |
| data.panel_outputs[].r_loop         | bool   | 轮播单是否循环     |
| data.panel_outputs[].r_src_id       | string | 轮播源 UUID    |
| data.panel_outputs[].r_src_uuid     | string | 轮播源 UUID    |
| data.panel_outputs[].r_src_progress | int    | 轮播源进度       |
| data.panel_outputs[].d_ip           | string | 设备 IP 地址    |
| data.other_inputs                   | array  | 其他输入列表      |
| data.panel_templates                | array  | 面板模板列表      |
| data.rotationlists                  | array  | 轮播列表        |

## 5.6.2 KLS 输出交叉输入

### 接口描述

KLS 面板输出输入交叉

### 请求地址

/v1/kls/panel/cross/output/input

### 请求方法

POST

### 请求参数示例

```
1 {
```

```
2  "panel_uuid": "panel-uuid-1",
3  "output_uuid": "output-uuid-1",
4  "input_name": "input1"
5 }
```

请求参数说明

| 参数名称        | 参数类型   | 是否必填 | 描述      |
|-------------|--------|------|---------|
| panel_uuid  | string | 是    | 面板 UUID |
| output_uuid | string | 是    | 输出 UUID |
| input_name  | string | 是    | 输入名称    |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

5.6.3 KLS 输入全选

接口描述

KLS 面板输入选择

请求地址

/v1/kls/panel/cross/input/select

请求方法

POST

请求参数示例

```
1 {
2   "panel_uuid": "panel-uuid-1",
3   "input_name": "input1",
4   "select": true
5 }
```

请求参数说明

| 参数名称       | 参数类型   | 是否必填 | 描述      |
|------------|--------|------|---------|
| panel_uuid | string | 是    | 面板 UUID |
| input_name | string | 是    | 输入名称    |
| select     | bool   | 是    | 是否全选    |

响应示例

```
1 {
2   "code": 200,
3   "message": "KLS 面板输入选择成功",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

6 KLS 绑定管理

6.1 绑定 KLS

接口描述

绑定 KLS

请求地址

/v1/bindkls/tobind

请求方法

POST

请求参数示例

```
1 {
2   "ip": "192.168.1.100",
3   "ifname": "1", // eth1.1
4   "location": 1
5 }
```

请求参数说明

| 参数名称     | 参数类型   | 是否必填 | 描述    |
|----------|--------|------|-------|
| ip       | string | 是    | IP 地址 |
| ifname   | string | 是    | 接口名称  |
| location | int    | 是    | 位置 ID |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

6.2 解绑 KLS

接口描述

解绑 KLS

请求地址

/v1/bindkls/unbind

请求方法

GET

请求参数示例

```
1 {
2     "id": 1
3 }
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| id   | int  | 是    | 位置 ID |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

6.3 查询 KLS VLAN

接口描述

获取 KLS VLAN

请求地址

/v1/bindkls/vlan

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "获取 KLS VLAN 成功",
4   "data": ["eth1.1"]
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | array  | 网卡名列表        |

6.4 获取绑定信息

接口描述

获取绑定信息

请求地址

/v1/bindkls/bindinfo

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "获取绑定信息成功",
4   "data": {
5     "status": 1,
6     "bind_list": [
7       {
8         "id": 1,
9         "ip": "192.168.1.100",
10        "ifname": "1", // eth1.1
11        "bind_status": 1
12      }
13    ]
14  }
15 }
```

响应参数说明

| 参数名称                         | 参数类型   | 描述                            |
|------------------------------|--------|-------------------------------|
| code                         | int    | 状态码，200 表示成功                  |
| message                      | string | 操作结果描述                        |
| data                         | object | 绑定信息                          |
| data.status                  | int    | 绑定状态 1:已绑定 2:绑定中 3:绑定失败 4:未绑定 |
| data.bind_list               | array  | 绑定列表                          |
| data.bind_list[].id          | int    | 位置 ID                         |
| data.bind_list[].ip          | string | IP 地址                         |
| data.bind_list[].ifname      | string | 接口名称                          |
| data.bind_list[].bind_status | int    | 绑定状态 1:已绑定 2:绑定中 3:绑定失败 4:未绑定 |

6.5 获取绑定状态

接口描述

获取绑定状态

请求地址

/v1/bindkls/status

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "status": 1
6   }
```

7 }

响应参数说明

| 参数名称        | 参数类型   | 描述                            |
|-------------|--------|-------------------------------|
| code        | int    | 状态码，200 表示成功                  |
| message     | string | 操作结果描述                        |
| data        | object | 绑定状态信息                        |
| data.status | int    | 绑定状态 1:已绑定 2:绑定中 3:绑定失败 4:未绑定 |

7 图标管理

7.1 上传图标

接口描述

上传图标

请求地址

/v1/icon/upload

请求方法

POST

请求参数示例

```
1 {
2   "index": 1,
3   "icon_file": "文件流",
4   "icon_name": "custom_icon.png",
5   "mime_type": "image/png",
6   "width": 64,
7   "height": 64
8 }
```

请求参数说明

| 参数名称      | 参数类型   | 是否必填 | 描述   |
|-----------|--------|------|------|
| index     | int    | 是    | 图标索引 |
| icon_file | file   | 是    | 图标文件 |
| icon_name | string | 否    | 图标名称 |

|           |        |   |         |
|-----------|--------|---|---------|
| mime_type | string | 否 | MIME 类型 |
| width     | int    | 否 | 图标宽度    |
| height    | int    | 否 | 图标高度    |

### 响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

## 7.2 修改图标

### 接口描述

修改自定义图标

### 请求地址

/v1/icon/modify

### 请求方法

POST

### 请求参数示例

```
1 {
2     "ids": [1, 2, 3]
3 }
```

### 请求参数说明

| 参数名称 | 参数类型  | 是否必填 | 描述       |
|------|-------|------|----------|
| ids  | array | 是    | 图标 ID 数组 |

### 响应示例

```
1 {
```

```
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

7.3 获取所有图标

接口描述

获取所有图标

请求地址

/v1/icon/all

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "official_icons": [
6             {
7                 "id": 1,
8                 "name": "icon1.png",
9                 "type": "official"
10            },
11            {
12                "id": 2,
13                "name": "icon2.png",
```

```
14         "type": "official"
15     },
16 ],
17     "custom_icons": [
18         {
19             "id": 3,
20             "name": "icon3.png",
21             "type": "custom"
22         }
23     ]
24 }
25 }
```

响应参数说明

| 参数名称                       | 参数类型   | 描述                   |
|----------------------------|--------|----------------------|
| code                       | int    | 状态码，200 表示成功         |
| message                    | string | 操作结果描述               |
| data                       | object | 图标列表                 |
| data.official_icons        | array  | 官方图标数组               |
| data.official_icons[].id   | int    | 图标 ID                |
| data.official_icons[].name | string | 图标名称                 |
| data.official_icons[].type | string | 图标类型：official/custom |
| data.custom_icons          | array  | 自定义图标数组              |
| data.custom_icons[].id     | int    | 图标 ID                |
| data.custom_icons[].name   | string | 图标名称                 |
| data.custom_icons[].type   | string | 图标类型：official/custom |

7.4 通知图标更新

接口描述

通知更新图标

请求地址

/v1/icon/notify/update

请求方法

POST

请求参数示例

```
1 {
2     "icon_name": "custom_icon.png"
3 }
```

请求参数说明

| 参数名称      | 参数类型   | 是否必填 | 描述   |
|-----------|--------|------|------|
| icon_name | string | 是    | 图标名称 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

8 前面板操作

8.1 输入预览

接口描述

输入预览

请求地址

/v1/front/input/preview

请求方法

POST

请求参数示例

```
1 {
2   "input_id": 1,
3   "enable": true
4 }
```

请求参数说明

| 参数名称     | 参数类型 | 是否必填 | 描述     |
|----------|------|------|--------|
| input_id | int  | 是    | 输入 ID  |
| enable   | bool | 是    | 是否启用预览 |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

8.2 执行切换

接口描述

输入输出切换

请求地址

/v1/front/input/output/take

请求方法

POST

请求参数示例

```
1 {
2     "panel_id": 1,
3     "input_id": 1,
4     "output_ids": [1, 2, 3]
5 }
```

请求参数说明

| 参数名称     | 参数类型 | 是否必填 | 描述     |
|----------|------|------|--------|
| panel_id | int  | 是    | 前面板 ID |

|            |       |   |          |
|------------|-------|---|----------|
| input_id   | int   | 是 | 输入 ID    |
| output_ids | array | 是 | 输出 ID 数组 |

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

8.3 查询前面板

接口描述

查询前面板

请求地址

/v1/front/panel/query

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "panels": [
```

```
6      {
7          "id": 1,
8          "name": "panel1",
9          "alias": "panel1"
10     },
11     {
12         "id": 2,
13         "name": "panel2",
14         "alias": "panel2"
15     }
16 ]
17 }
18 }
```

响应参数说明

| 参数名称                | 参数类型   | 描述           |
|---------------------|--------|--------------|
| code                | int    | 状态码，200 表示成功 |
| message             | string | 操作结果描述       |
| data                | object | 数据对象         |
| data.panels         | array  | 面板数组         |
| data.panels[].id    | int    | 面板 ID        |
| data.panels[].name  | string | 面板名称         |
| data.panels[].alias | string | 别名名称         |

8.4 获取前面板详情

接口描述

获取前面板详情

请求地址

/v1/front/panel/detail

请求方法

GET

请求参数示例

```
1 {
2     "panel_id": 1
3 }
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述 |
|------|------|------|----|
|------|------|------|----|

|          |     |   |        |
|----------|-----|---|--------|
| panel_id | int | 是 | 前面板 ID |
|----------|-----|---|--------|

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "id": 1,
6     "name": "panel1",
7     "alias": "panel1",
8     "panel_inputs": [
9       {
10        "id": 1,
11        "name": "input1",
12        "alias": "input1",
13        "icon_name": "input1",
14        "front_content": "input1"
15      },
16      {
17        "id": 2,
18        "name": "input2",
19        "alias": "input2",
20        "icon_name": "input2",
21        "front_content": "input2"
22      }
23    ],
24    "panel_outputs": [
25      {
26        "id": 1,
27        "name": "output1",
28        "alias": "output1",
29        "lock": false,
30        "input_id": 1,
31        "linked": false,
32        "icon_name": "output1",
33        "front_content": "output1"
34      }
35    ]
36  }
37 }
```

响应参数说明

| 参数名称                   | 参数类型   | 描述           |
|------------------------|--------|--------------|
| code                   | int    | 状态码，200 表示成功 |
| message                | string | 操作结果描述       |
| data                   | object | 数据对象         |
| data.id                | int    | 面板 ID        |
| data.name              | string | 面板名称         |
| data.alias             | string | 别名名称         |
| data.panel_inputs      | array  | 输入数组         |
| data.panel_inputs[].id | int    | 输入 ID        |

|                                    |        |        |
|------------------------------------|--------|--------|
| data.panel_inputs[].name           | string | 输入名称   |
| data.panel_inputs[].alias          | string | 输入别名   |
| data.panel_inputs[].icon_name      | string | 输入图标名  |
| data.panel_inputs[].front_content  | string | 输入前端内容 |
| data.panel_outputs                 | array  | 输出数组   |
| data.panel_outputs[].id            | int    | 输出 ID  |
| data.panel_outputs[].name          | string | 输出名称   |
| data.panel_outputs[].alias         | string | 输出别名   |
| data.panel_outputs[].lock          | bool   | 是否锁定输出 |
| data.panel_outputs[].input_id      | int    | 输入 ID  |
| data.panel_outputs[].linked        | bool   | 是否链接输入 |
| data.panel_outputs[].icon_name     | string | 输出图标名  |
| data.panel_outputs[].front_content | string | 输出前端内容 |

## 8.5 停止 NDI

### 接口描述

停止 NDI

### 请求地址

/v1/front/ndi/stop

### 请求方法

POST

### 请求参数示例

无

### 请求参数说明

无

### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

9 VLAN 配置

9.1 获取所有 VLAN 配置

接口描述

获取所有 VLAN 配置

请求地址

/v1/vlan/cfg/all

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "vlan_cfgs": [
6       {
7         "id": 1,
8         "type": true,
9         "name": "NDI",
10        "priority": "Highest",
11        "description": "Default VLAN NDI",
12        "vlans": []
13      }
14    ]
15  }
```

```
15     }
16 }
```

## 响应参数说明

| 参数名称             | 参数类型   | 描述           |
|------------------|--------|--------------|
| code             | int    | 状态码，200 表示成功 |
| message          | string | 操作结果描述       |
| data             | object | 数据对象         |
| data.id          | int    | 面板 ID        |
| data.name        | string | 面板名称         |
| data.type        | bool   | 类型（是否为官方）    |
| data.priority    | string | 优先级          |
| data.description | string | 描述           |
| data.vlans       | array  | vlan 列表      |

## 9.2 修改 VLAN 配置

### 接口描述

修改 VLAN 配置

### 请求地址

/v1/vlan/cfg/modify

### 请求方法

POST

### 请求参数示例

```
1 {
2     "id": 1,
3     "vlan_ids": "1"
4 }
```

### 请求参数说明

| 参数名称     | 参数类型  | 是否必填 | 描述         |
|----------|-------|------|------------|
| id       | int   | True | ID         |
| vlan_ids | array | True | vlan ID 列表 |

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

10 拓扑管理

10.1 查询拓扑结构

接口描述

查询拓扑结构

请求地址

/v1/topology/query

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "nodes": [
6       {
```

```

7      "id": "18:C3:F4:2F:FB:EE",
8      "name": "AVX24-4-chen666",
9      "type": "kiloview",
10     "label": "AVX24-4",
11     "parent_id": "",
12     "shape": "router",
13     "image": "./static/img/AVX24_4.png",
14     "valid": false,
15     "x": 320,
16     "y": 560,
17     "connection_port": 0
18   },
19   {
20     "id": "V:18:C3:F4:2F:FB:EE-3",
21     "name": "unknownSwitch",
22     "type": "other",
23     "label": "unknown",
24     "parent_id": "18:C3:F4:2F:FB:EE",
25     "shape": "device",
26     "image": "./static/img/unknown_switch.png",
27     "valid": false,
28     "x": 325,
29     "y": 690,
30     "connection_port": 0
31   },
32   {
33     "id": "18:C3:F4:2F:FB:F3",
34     "name": "AVX24-4-lion",
35     "type": "kiloview",
36     "label": "AVX24-4",
37     "parent_id": "V:18:C3:F4:2F:FB:EE-3",
38     "shape": "routerTerminal",
39     "image": "./static/img/AVX24_4.png",
40     "valid": false,
41     "x": 175,
42     "y": 820,
43     "connection_port": 1
44   },
45   {
46     "id": "18:C3:F4:2F:FB:E7",
47     "name": "AVX24-4",
48     "type": "kiloview",
49     "label": "AVX24-4",
50     "parent_id": "V:18:C3:F4:2F:FB:EE-3",
51     "shape": "routerTerminal",
52     "image": "./static/img/AVX24_4.png",
53     "valid": false,
54     "x": 325,
55     "y": 820,
56     "connection_port": 1
57   },
58   {
59     "id": "10:78:56:34:12:61",
60     "name": "AVX24-4",
61     "type": "kiloview",
62     "label": "AVX24-4",
63     "parent_id": "V:18:C3:F4:2F:FB:EE-3",
64     "shape": "routerTerminal",
65     "image": "./static/img/AVX24_4.png",
66     "valid": false,

```

```

67         "x": 475,
68         "y": 820,
69         "connection_port": 19
70     },
71     {
72         "id": "18:C3:F4:25:6A:D2",
73         "name": "unknown",
74         "type": "other",
75         "label": "unknown",
76         "parent_id": "18:C3:F4:2F:FB:EE",
77         "shape": "device",
78         "image": "./static/img/unknown.png",
79         "valid": false,
80         "x": 625,
81         "y": 690,
82         "connection_port": 0
83     }
84 ],
85 "wires": [
86     {
87         "id": "18:C3:F4:2F:FB:EE:3",
88         "shape": "edge",
89         "color": ""
90     },
91     {
92         "id": "V:18:C3:F4:2F:FB:EE-3:35",
93         "shape": "edge",
94         "color": ""
95     },
96     {
97         "id": "V:18:C3:F4:2F:FB:EE-3:42",
98         "shape": "edge",
99         "color": ""
100    },
101    {
102        "id": "V:18:C3:F4:2F:FB:EE-3:85",
103        "shape": "edge",
104        "color": ""
105    },
106    {
107        "id": "18:C3:F4:2F:FB:EE:11",
108        "shape": "edge",
109        "color": ""
110    }
111 ]
112 }
113 }

```

## 响应参数说明

| 参数名称       | 参数类型   | 描述           |
|------------|--------|--------------|
| code       | int    | 状态码，200 表示成功 |
| message    | string | 操作结果描述       |
| data       | object | 返回数据         |
| data.nodes | array  | 节点列表         |

|                               |        |         |
|-------------------------------|--------|---------|
| data.nodes[ ].id              | string | 节点 ID   |
| data.nodes[ ].name            | string | 节点名称    |
| data.nodes[ ].type            | string | 节点类型    |
| data.nodes[ ].label           | string | 节点标签    |
| data.nodes[ ].parent_id       | string | 父节点 ID  |
| data.nodes[ ].shape           | string | 节点形状    |
| data.nodes[ ].image           | string | 节点图片    |
| data.nodes[ ].valid           | bool   | 节点是否有效  |
| data.nodes[ ].x               | int    | 节点 X 坐标 |
| data.nodes[ ].y               | int    | 节点 Y 坐标 |
| data.nodes[ ].connection_port | int    | 节点连接端口  |
| data.wires                    | array  | 线路列表    |
| data.wires[ ].id              | string | 线路 ID   |
| data.wires[ ].shape           | string | 线路形状    |
| data.wires[ ].color           | string | 线路颜色    |

## 10.2 查询拓扑状态

### 接口描述

查询拓扑状态

### 请求地址

/v1/topology/status/query

### 请求方法

GET

### 请求参数示例

无

### 请求参数说明

无

### 响应示例

```
1 {
```

```

2  "data": {
3      "nodes": [
4          {
5              "id": "18:C3:F4:25:6A:D2",
6              "isMain": false,
7              "status": "online",
8              "alarm_status": "",
9
10             "ports": [ ]
11         },
12         {
13             "id": "10:78:56:34:12:61",
14             "isMain": false,
15             "status": "online",
16             "alarm_status": "",
17
18             "ports": [ ]
19         },
20         {
21             "id": "18:C3:F4:2F:FB:F3",
22             "isMain": false,
23             "status": "online",
24             "alarm_status": "",
25
26             "ports": [ ]
27         },
28         {
29             "id": "18:C3:F4:2F:FB:E7",
30             "isMain": false,
31             "status": "online",
32             "alarm_status": "",
33
34             "ports": [ ]
35         },
36         {
37             "id": "V:18:C3:F4:2F:FB:EE-3",
38             "isMain": false,
39             "status": "online",
40             "alarm_status": "",
41
42             "ports": [ ]
43         },
44         {
45             "id": "18:C3:F4:2F:FB:EE",
46             "isMain": true,
47             "status": "online",
48             "alarm_status": "",
49             "ports": [
50                 {
51                     "port_id": "3",
52                     "status": "online"
53                 },
54                 {
55                     "port_id": "11",
56                     "status": "online"
57                 }
58             ]
59         }
60     ]
61 }

```

```

62     }
63   ]
64 }
65 ],
66 "edges": [
67   {
68     "id": "18:C3:F4:2F:FB:EE:11",
69     "fill": "",
70     "status": "online",
71     "source": {
72       "is_dev": false,
73       "cell": "18:C3:F4:2F:FB:EE",
74       "port": "11"
75     },
76     "target": {
77       "is_dev": true,
78       "cell": "18:C3:F4:25:6A:D2",
79       "port": ""
80     }
81   },
82   {
83     "id": "18:C3:F4:2F:FB:EE:3",
84     "fill": "",
85     "status": "online",
86     "source": {
87       "is_dev": false,
88       "cell": "18:C3:F4:2F:FB:EE",
89       "port": "3"
90     },
91     "target": {
92       "is_dev": true,
93       "cell": "V:18:C3:F4:2F:FB:EE-3",
94       "port": ""
95     }
96   },
97   {
98     "id": "V:18:C3:F4:2F:FB:EE-3:85",
99     "fill": "",
100    "status": "online",
101    "source": {
102      "is_dev": false,
103      "cell": "V:18:C3:F4:2F:FB:EE-3",
104      "port": "85"
105    },
106    "target": {
107      "is_dev": true,
108      "cell": "10:78:56:34:12:61",
109      "port": ""
110    }
111  },
112  {
113    "id": "V:18:C3:F4:2F:FB:EE-3:35",
114    "fill": "",
115    "status": "online",
116    "source": {
117      "is_dev": false,
118      "cell": "V:18:C3:F4:2F:FB:EE-3",
119      "port": "35"
120    },
121    "target": {

```

```

122         "is_dev": true,
123         "cell": "18:C3:F4:2F:FB:F3",
124         "port": ""
125     },
126 },
127 {
128     "id": "V:18:C3:F4:2F:FB:EE-3:42",
129     "fill": "",
130     "status": "online",
131     "source": {
132         "is_dev": false,
133         "cell": "V:18:C3:F4:2F:FB:EE-3",
134         "port": "42"
135     },
136     "target": {
137         "is_dev": true,
138         "cell": "18:C3:F4:2F:FB:E7",
139         "port": ""
140     }
141 }
142 ]
143 }
144 }

```

## 响应参数说明

| 参数名称                           | 参数类型   | 描述           |
|--------------------------------|--------|--------------|
| code                           | int    | 状态码，200 表示成功 |
| message                        | string | 操作结果描述       |
| data                           | object | 返回数据         |
| data.nodes                     | array  | 节点列表         |
| data.nodes[ ].id               | string | 节点 ID        |
| data.nodes[ ].isMain           | bool   | 是否为主节点       |
| data.nodes[ ].status           | string | 节点状态         |
| data.nodes[ ].alarm_status     | string | 报警状态         |
| data.nodes[ ].ports            | array  | 端口列表         |
| data.nodes[ ].ports[ ].port_id | string | 端口 ID        |
| data.nodes[ ].ports[ ].status  | string | 端口状态         |
| data.wires                     | array  | 线路列表         |
| data.wires[ ].id               | string | 线路 ID        |
| data.wires[ ].fill             | string | 线路填充         |
| data.wires[ ].status           | string | 线路状态         |
| data.wires[ ].source           | object | 线路源          |
| data.wires[ ].source.is_dev    | bool   | 是否为设备        |
| data.wires[ ].source.cell      | string | 源设备 ID       |
| data.wires[ ].source.port      | string | 源端口          |
| data.wires[ ].target           | object | 线路目标         |

|                             |        |         |
|-----------------------------|--------|---------|
| data.wires[ ].target.is_dev | bool   | 是否为设备   |
| data.wires[ ].target.is_dev | bool   | 是否为设备   |
| data.wires[ ].target.cell   | string | 目标设备 ID |
| data.wires[ ].target.port   | string | 目标端口    |

### 10.3 查询设备信息

#### 接口描述

查询拓扑设备的详情。

#### 请求地址

/v1/topology/dev/info/query

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "data": {
3         "port_id": 13,
4         "dev_mac": ""
5     }
6 }
7
8 或者
9 {
10    "data": {
11        "port_id": 0,
12        "dev_mac": "18:C3:F4:20:64:E1"
13    }
14 }
```

#### 请求参数说明

以下仅说明请求体字段（不含 HTTP 头）。

#### 说明约定

- 1. **data** 为对象类型（对应 topologyDeviceReq），其子字段单独见下表。
- 2. **port\_id** 与 **dev\_mac**：业务上通常只使用其一作为查询键——按端口查时填有效 port\_id、dev\_mac 可传空字符串；按 MAC 查时填有效 dev\_mac、port\_id 传 0。

#### data (topologyDeviceReq) 对象字段说明 (data.port\_id、data.dev\_mac)

| 参数名称 | 参数类型 | 是否必填 | 描述 |
|------|------|------|----|
|------|------|------|----|

|         |        |      |                                           |
|---------|--------|------|-------------------------------------------|
| port_id | Number | True | 交换机端口 ID; 按 MAC 查询时传 0                    |
| dev_mac | String | True | 设备 MAC (大写/小写以网关实现为准); 按端口查询时可为空字符串<br>"" |

### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "switch_id": 0,
6     "port_id": 13,
7     "dev_name": "18:C3:F4:20:64:E1",
8     "dev_mac": "18:C3:F4:20:64:E1",
9     "dev_ip": "192.168.3.225",
10    "is_static_ip": false,
11    "image": "./static/img/N50.png",
12    "rx": {
13      "packets": "2",
14      "bytes": "1203",
15      "error": "0",
16      "loss": "0"
17    },
18    "tx": {
19      "packets": "1",
20      "bytes": "491",
21      "error": "0",
22      "loss": "0"
23    },
24    "media": [],
25    "model": "N50",
26    "sn": "32021120064E1",
27    "software_version": "2.43.0011.kdds.0297",
28    "hardware_version": "",
29    "company": "Kiloview",
30    "alias": "N50-32021120064E1",
31    "sw_serial_number": "18c3f425640e",
32    "diagnose_status": 0,
33    "parent_id": "",
34    "cpu_usage": "16.20%",
35    "disk_usage": "48.75%",
36    "mem_usage": "21.44%",
37    "temperature": "",
38    "server_count": 1,
39    "dev_proxy_url": "",
40    "switch_link_status": "online"
41  }
42 }
```

### 响应参数说明

顶层 code、message 与其它接口一致：code 为业务码（如 200 表示成功），message 为说明字符串。

以下仅说明 **data (topologyDevInfo)** 及其嵌套对象字段。

### 说明约定

1. **data 中的 Object 字段** (rx、tx)：先在本节用独立表格列出子字段；末尾「data 对象字段说明」表中对应行写「见 xxx」，不重复展开。
2. **media**：为对象数组，元素类型为 DevMediaInfo，单独用「media 数组元素字段说明」描述。
3. **rx / tx**：结构相同（均为 SwitchPortRate），共用一张「data.rx / data.tx 对象字段说明」。

### data.rx / data.tx 对象字段说明 (data.rx、data.tx，类型均为 SwitchPortRate)

| 参数名称    | 参数类型            | 描述                           |
|---------|-----------------|------------------------------|
| packets | Number / String | 报文个数（序列化实现不同可能为 JSON 数字或字符串） |
| bytes   | Number / String | 字节数                          |
| error   | Number / String | 错误报文数                        |
| loss    | Number / String | 丢弃报文数                        |

### media 数组元素字段说明 (data.media[], 类型 DevMediaInfo)

| 参数名称           | 参数类型   | 描述         |
|----------------|--------|------------|
| ndi_src        | String | NDI 源标识    |
| url            | String | 媒体 URL 或地址 |
| video_codec    | String | 视频编码格式     |
| height         | Number | 视频高度（像素）   |
| width          | Number | 视频宽度（像素）   |
| audio_codec    | String | 音频编码格式     |
| sampling       | Number | 采样率相关参数    |
| channel_number | Number | 声道数        |
| bitrate        | Number | 码率（浮点）     |
| framerate      | Number | 帧率         |

### data 对象字段说明 (topologyDevInfo)

| 参数名称         | 参数类型    | 描述                                   |
|--------------|---------|--------------------------------------|
| switch_id    | Number  | 交换机标识；单机场景多为 0                       |
| port_id      | Number  | 设备所在端口 ID                            |
| dev_name     | String  | 拓扑/设备名称（常与 MAC 或展示名相关）               |
| dev_mac      | String  | 设备 MAC                               |
| dev_ip       | String  | 设备 IP                                |
| is_static_ip | Boolean | 是否为静态 IP                             |
| image        | String  | 设备图标或图片路径                            |
| rx           | Object  | 接收方向流量统计，见「data.rx / data.tx 对象字段说明」 |

|                    |        |                                      |
|--------------------|--------|--------------------------------------|
| tx                 | Object | 发送方向流量统计，见「data.rx / data.tx 对象字段说明」 |
| media              | Array  | 媒体信息列表，元素见「media 数组元素字段说明」           |
| model              | String | 设备型号                                 |
| sn                 | String | 设备序列号                                |
| software_version   | String | 软件版本                                 |
| hardware_version   | String | 硬件版本                                 |
| company            | String | 厂商                                   |
| alias              | String | 设备别名（展示用）                            |
| sw_serial_number   | String | 本交换机序列号或关联网关序列号（以实际返回为准）             |
| diagnose_status    | Number | 诊断状态：0 无诊断，1 诊断中                     |
| parent_id          | String | 父节点或拓扑关联 ID；无则为空字符串                  |
| cpu_usage          | String | CPU 占用率展示字符串，如 16.20%                |
| disk_usage         | String | 磁盘占用率展示字符串                           |
| mem_usage          | String | 内存占用率展示字符串                           |
| temperature        | String | 温度展示字符串；无数据时可为空字符串                   |
| server_count       | Number | 同时上报的服务端个数                           |
| dev_proxy_url      | String | 反代访问设备 Web 的 URL；无则为空字符串             |
| switch_link_status | String | 与交换机的链路状态，如 online、offline 等         |

## 11 网络诊断

### 11.1 诊断管理

#### 11.1.1 KLS 跨设备开启端到端诊断

##### 接口描述

开启 KLS 端到端诊断任务（KLS 下发到 AVX）

##### 请求地址

/v1/diagnose/task/rfppadd

## 请求方法

POST

## 请求参数示例

仅必填参数（使用默认值）示例：

```
1 {  
2   "task_id": 1001,  
3   "type": 0,  
4   "from_dev": "AVX-001",  
5   "to_dev": "DEV-002",  
6   "from_mac": "00:11:22:33:44:55",  
7   "to_mac": "66:77:88:99:AA:BB",  
8   "from_ip": "192.168.1.10",  
9   "to_ip": "192.168.1.20",  
10  "from_type": 0,  
11  "to_type": 1  
12 }
```

全参数配置示例：

```
1 {  
2   "task_id": 1001,  
3   "type": 2,  
4   "role": "1222333",  
5   "packet_len": 1400,  
6   "speed": 900,  
7   "duration": 120,  
8   "from_dev": "AVX-001",  
9   "to_dev": "DEV-002",  
10  "from_mac": "00:11:22:33:44:55",  
11  "to_mac": "66:77:88:99:AA:BB",  
12  "from_ip": "192.168.1.10",  
13  "to_ip": "192.168.1.20",  
14  "from_type": 0,  
15  "to_type": 1  
16 }
```

## 请求参数说明

| 参数名称       | 参数类型   | 是否必填  | 描述                           |
|------------|--------|-------|------------------------------|
| task_id    | int32  | True  | KLS 任务 ID                    |
| type       | int32  | True  | 诊断类型。枚举：0 - 连通，1 - 快速，2 - 性能 |
| role       | string | False | 填入 RF02SN                    |
| packet_len | int32  | False | 包长，默认 1400                   |
| speed      | int32  | False | 带宽，单位 Mbps，默认 900            |
| duration   | int32  | False | 诊断时长，单位秒，默认 120              |
| from_dev   | string | True  | 源设备名称                        |

|           |        |      |                               |
|-----------|--------|------|-------------------------------|
| to_dev    | string | True | 目标设备名称                        |
| from_mac  | string | True | 源设备 MAC 地址                    |
| to_mac    | string | True | 目标设备 MAC 地址                   |
| from_ip   | string | True | 源设备 IP 地址                     |
| to_ip     | string | True | 目标设备 IP 地址                    |
| from_type | int32  | True | 源设备类型。枚举：<br>0-AVX, 1 - 普通设备  |
| to_type   | int32  | True | 目标设备类型。枚举：<br>0-AVX, 1 - 普通设备 |

### 响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {}
5 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

## 11.1.2 获取当前交换机所有设备列表

### 接口描述

获取当前交换机所有设备列表（端到端诊断专用）。

### 请求地址

/v1/diagnose/task/alldev/list

### 请求方法

GET

### 请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "name": "AVX-001",
7       "mac": "00:11:22:33:44:55",
8       "serial_number": "SN20240001",
9       "ip": "192.168.1.10"
10    }
11  ]
12 }
```

响应参数说明

| 参数名称                 | 参数类型   | 描述             |
|----------------------|--------|----------------|
| code                 | int32  | 业务状态码，200 表示成功 |
| message              | string | 业务提示信息         |
| data                 | array  | 设备列表           |
| data[].name          | string | 设备名称           |
| data[].mac           | string | 设备 MAC 地址      |
| data[].serial_number | string | 设备序列号          |
| data[].ip            | string | 设备 IP 地址       |

11.1.3 端诊断任务删除

接口描述

删除指定的诊断任务，支持批量删除。

请求地址

/v1/diagnose/task/del

请求方法

POST

请求参数示例

```
1 {
2   "ids": [2001, 2002, 2003]
```

3 }

请求参数说明

| 参数名称 | 参数类型  | 是否必填 | 描述         |
|------|-------|------|------------|
| ids  | array | True | 诊断任务 ID 数组 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {}
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | object | 返回数据，可能为空      |

11.1.4 诊断设备结果查询

接口描述

查询诊断设备结果，用于诊断进度轮询。

请求地址

/v1/diagnose/task/device

请求方法

POST

请求参数示例

```
1 {
2     "task_id": 2001,
3     "task_type": 0
4 }
```

请求参数说明

| 参数名称    | 参数类型  | 是否必填 | 描述      |
|---------|-------|------|---------|
| task_id | int32 | True | 诊断任务 ID |

|           |       |      |                             |
|-----------|-------|------|-----------------------------|
| task_type | int32 | True | 任务类型。枚举：0 - 全局诊断, 1 - 端到端诊断 |
|-----------|-------|------|-----------------------------|

## 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "total": 5,
5   "data": {
6     "result_msg": "诊断完成",
7     "device_info_list": [
8       {
9         "id": 1,
10        "task_id": 2001,
11        "result_type": 0,
12        "from_dev": "AVX-001",
13        "to_dev": "DEV-002",
14        "ping_result": 1,
15        "delay": "5ms",
16        "jitter": "1ms",
17        "loss": "0.1%",
18        "rx": "950Mbps",
19        "tx": "940Mbps",
20        "ip": "192.168.1.20",
21        "devmac": "66:77:88:99:AA:BB",
22        "switcher_sn": "SN20240001",
23        "create_time": "2024-01-01 10:00:00",
24        "update_time": "2024-01-01 10:03:00",
25        "check_device_data": {
26          "check_delay": true,
27          "check_loss": true,
28          "check_tx": true,
29          "check_device": true,
30          "check_ping": true
31        },
32        "total_info": "1000/1",
33        "dev_type": "普通设备",
34        "identify": 1
35      }
36    ],
37    "device_count": 5,
38    "done": 100,
39    "diagnose_time": "2024-01-01 10:03:00",
40    "all_er": {
41      "er_delay": 0,
42      "er_loss": 0,
43      "er_tx": 0,
44      "er_device": 0,
45      "er_ping": 0
46    },
47    "normal": true,
48    "diagnose_info": {
49      "abnormal": 0,
50      "dtime": "180s",
51      "from_dev": "AVX-001",
52      "from_ip": "192.168.1.10",
53      "id": 1,
```

```

54         "mac_process": 100,
55         "macping_test": 1,
56         "normal": 5,
57         "ping_process": 100,
58         "ping_test": 1,
59         "recognize": 5,
60         "unrecognized": 0,
61         "step": 2,
62         "task_id": 2001,
63         "to_dev": "DEV-002",
64         "to_ip": "192.168.1.20",
65         "network_diagnose": 1
66     }
67 }
68 }

```

## 响应参数说明

| 参数名称                                                  | 参数类型   | 描述                                            |
|-------------------------------------------------------|--------|-----------------------------------------------|
| code                                                  | int32  | 业务状态码，200 表示成功                                |
| message                                               | string | 业务提示信息                                        |
| total                                                 | int32  | 诊断设备总数量                                       |
| data                                                  | object | 诊断结果核心数据                                      |
| data.result_msg                                       | string | 诊断结果文本                                        |
| data.device_info_list                                 | array  | 设备诊断列表                                        |
| data.device_info_list[].id                            | int32  | 记录 ID                                         |
| data.device_info_list[].task_id                       | int32  | 任务 ID                                         |
| data.device_info_list[].result_type                   | int32  | 结果类型                                          |
| data.device_info_list[].from_dev                      | string | 源设备名称                                         |
| data.device_info_list[].to_dev                        | string | 目标设备名称                                        |
| data.device_info_list[].ping_result                   | int32  | Ping 状态：0 - 未连通，1 - 连通，2 - 未检测（新版本仅端到端才 Ping） |
| data.device_info_list[].delay                         | string | Ping 时延                                       |
| data.device_info_list[].jitter                        | string | 抖动（暂时不用）                                      |
| data.device_info_list[].loss                          | string | 丢包率                                           |
| data.device_info_list[].rx                            | string | 平均速率                                          |
| data.device_info_list[].tx                            | string | 接收速率（暂时不用）                                    |
| data.device_info_list[].ip                            | string | 设备 IP 地址                                      |
| data.device_info_list[].devmac                        | string | 设备 MAC 地址                                     |
| data.device_info_list[].switcher_sn                   | string | 交换机序列号                                        |
| data.device_info_list[].create_time                   | string | 创建时间                                          |
| data.device_info_list[].update_time                   | string | 更新时间                                          |
| data.device_info_list[].check_device_data             | object | 诊断校验结果                                        |
| data.device_info_list[].check_device_data.check_delay | bool   | 时延是否正常                                        |
| data.device_info_list[].check_device_data.check_loss  | bool   | 丢包是否正常                                        |

|                                                        |        |                                              |
|--------------------------------------------------------|--------|----------------------------------------------|
| data.device_info_list[].check_device_data.check_tx     | bool   | 发送是否正常                                       |
| data.device_info_list[].check_device_data.check_device | bool   | 设备是否正常                                       |
| data.device_info_list[].check_device_data.check_ping   | bool   | Ping 是否正常                                    |
| data.device_info_list[].total_info                     | string | 总包数 / 丢包数信息（已废弃）                             |
| data.device_info_list[].dev_type                       | string | 设备类型                                         |
| data.device_info_list[].identify                       | int32  | 是否识别: 0 - 未识别, 1 - 识别                        |
| data.device_count                                      | int32  | 诊断设备总数（目前不用）                                 |
| data.done                                              | int32  | 诊断进度条                                        |
| data.diagnose_time                                     | string | 诊断时间                                         |
| data.all_er                                            | object | 异常统计（目前不用）                                   |
| data.all_er.er_delay                                   | int32  | 时延异常数量                                       |
| data.all_er.er_loss                                    | int32  | 丢包异常数量                                       |
| data.all_er.er_tx                                      | int32  | 速率异常数量                                       |
| data.all_er.er_device                                  | int32  | 设备异常数量                                       |
| data.all_er.er_ping                                    | int32  | Ping 异常数量                                    |
| data.normal                                            | bool   | 当前诊断结果是否正常（目前不用）                             |
| data.process                                           | object | 诊断流程（已废弃）                                    |
| data.diagnose_info                                     | object | 新版全局诊断信息                                     |
| data.diagnose_info.abnormal                            | int32  | 异常设备数量（全局诊断）                                 |
| data.diagnose_info.dtime                               | string | 诊断时间                                         |
| data.diagnose_info.from_dev                            | string | 源设备名称                                        |
| data.diagnose_info.from_ip                             | string | 源设备 IP 地址                                    |
| data.diagnose_info.id                                  | int32  | 诊断信息 ID                                      |
| data.diagnose_info.mac_process                         | int32  | 网络连通性测试进度                                    |
| data.diagnose_info.macping_test                        | int32  | 网络连通性测试状态: 0 - 测试中, 1 - 成功, 2 - 失败, 3 - 未测试  |
| data.diagnose_info.normal                              | int32  | 正常设备数量（全局诊断）                                 |
| data.diagnose_info.ping_process                        | int32  | IP 连通性测试进度                                   |
| data.diagnose_info.ping_test                           | int32  | IP 连通性测试状态: 0 - 测试中, 1 - 成功, 2 - 失败, 3 - 未测试 |
| data.diagnose_info.recognize                           | int32  | 识别设备数量（全局诊断）                                 |
| data.diagnose_info.unrecognized                        | int32  | 未识别设备数量（全局诊断）                                |
| data.diagnose_info.step                                | int32  | 当前进度（端到端 / 全局均两步）                            |
| data.diagnose_info.task_id                             | int32  | 事项 ID                                        |

|                                     |        |                 |
|-------------------------------------|--------|-----------------|
| data.diagnose_info.to_dev           | string | 目标设备名称          |
| data.diagnose_info.to_ip            | string | 目标设备 IP 地址      |
| data.diagnose_info.network_diagnose | int32  | 网络诊断是否完成：1 - 完成 |

## 11.1.5 开启端到端诊断

### 接口描述

开启通用端到端诊断任务。

### 请求地址

/v1/diagnose/task/p2pdev/add

### 请求方法

POST

### 请求参数示例

仅必填参数（使用默认值）示例：

```
1 {
2   "from_dev": "AVX-001",
3   "from_ip": "192.168.1.10",
4   "from_mac": "00:11:22:33:44:55",
5   "to_dev": "DEV-002",
6   "to_ip": "192.168.1.20",
7   "to_mac": "66:77:88:99:AA:BB",
8   "type": 0
9 }
```

全参数配置示例：

```
1 {
2   "from_dev": "AVX-001",
3   "from_ip": "192.168.1.10",
4   "from_mac": "00:11:22:33:44:55",
5   "to_dev": "DEV-002",
6   "to_ip": "192.168.1.20",
7   "to_mac": "66:77:88:99:AA:BB",
8   "type": 2,
9   "role": "sendrecv",
10  "speed": 1000,
11  "duration": 180,
12  "packet_len": 1500
13 }
```

### 请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述 |
|------|------|------|----|
|------|------|------|----|

|            |        |       |                              |
|------------|--------|-------|------------------------------|
| from_dev   | string | True  | 设备 1 名称（源设备）                 |
| from_ip    | string | True  | 设备 1 IP 地址                   |
| from_mac   | string | True  | 设备 1 MAC 地址                  |
| to_dev     | string | True  | 设备 2 名称（目标设备）                |
| to_ip      | string | True  | 设备 2 IP 地址                   |
| to_mac     | string | True  | 设备 2 MAC 地址                  |
| type       | int32  | True  | 诊断类型。枚举：0 - 连通，1 - 快速，2 - 性能 |
| role       | string | False | 诊断收发方式。枚举：recv/send/sendrecv |
| speed      | int32  | False | 带宽，单位 Mbps                   |
| duration   | int32  | False | 诊断时长，单位秒                     |
| packet_len | int32  | False | 包长                           |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {
5         "task_id": 3001
6     }
7 }
```

响应参数说明

| 参数名称         | 参数类型   | 描述             |
|--------------|--------|----------------|
| code         | int32  | 业务状态码，200 表示成功 |
| message      | string | 业务提示信息         |
| data         | object | 返回数据           |
| data.task_id | int32  | 诊断任务 ID        |

11.1.6 查询诊断任务（按类型）

接口描述

按任务类型查询历史诊断任务列表。

请求地址

/v1/diagnose/task/query/type

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "id": 3001,
7       "name": "端到端诊断任务",
8       "type": 1,
9       "from_dev": "AVX-001",
10      "to_dev": "DEV-002",
11      "seq": 1,
12      "speed": 1000,
13      "duration": 180,
14      "pkg_len": 1500,
15      "vlan_id": 100,
16      "status": 1,
17      "switcher_sn": "SN20240001",
18      "create_time": "2024-01-01 10:00:00",
19      "update_time": "2024-01-01 10:03:00"
20    }
21  ]
22 }
```

响应参数说明

| 参数名称            | 参数类型   | 描述                         |
|-----------------|--------|----------------------------|
| code            | int32  | 业务状态码，200 表示成功             |
| message         | string | 业务提示信息                     |
| data            | array  | 诊断任务列表                     |
| data[].id       | int32  | 任务 ID                      |
| data[].name     | string | 任务名称                       |
| data[].type     | int32  | 任务类型。枚举：0 - 全局诊断，1 - 端到端诊断 |
| data[].from_dev | string | 源设备名称                      |
| data[].to_dev   | string | 目标设备名称                     |
| data[].seq      | int32  | 序列号                        |

|                    |        |                                                |
|--------------------|--------|------------------------------------------------|
| data[].speed       | int32  | 测速速率, 单位 Mbps                                  |
| data[].duration    | int32  | 诊断时长, 单位秒                                      |
| data[].pkg_len     | int32  | 包长                                             |
| data[].vlan_id     | int32  | VLAN ID                                        |
| data[].status      | int32  | 任务状态。枚举: 0 - 进行中, 1 - 完成, 2 - 任务部分失败, 3 - 任务失败 |
| data[].switcher_sn | string | 交换机序列号                                         |
| data[].create_time | string | 创建时间                                           |
| data[].update_time | string | 更新时间                                           |

## 12 交换机管理

### 12.1 系统信息

#### 12.1.1 查询交换机系统信息

##### 接口描述

获取交换机基本信息、设备信息、型号、别名、序列号、KiloLink 连接状态等。

##### 请求地址

/v1/switch/sys/info

##### 请求方法

GET

##### 请求参数示例

无

##### 请求参数说明

无

##### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
```

```
5      "basic_info": {
6          "loc_mac": "18:C3:F4:2F:FB:F3",
7          "uptime": "1d 07:10:14",
8          "soft_version": "0.03.0268",
9          "hw_version": "V0.0-V1-0.0-0.0"
10     },
11     "dev_info": {
12         "cpu_usage": "10%",
13         "mem_usage": "12%",
14         "temperature": "73",
15         "fan": "87%"
16     },
17     "model": "AVX24-4",
18     "alias": "AVX24-4",
19     "sn": "18c3f425640e",
20     "kls_link_status": "disconnected",
21     "hostname": "AVX24-4-lion",
22     "used_port": 6,
23     "total_port": 28,
24     "mode": "layer3"
25 }
26 }
```

响应参数说明

| 参数名称            | 参数类型   | 描述              |
|-----------------|--------|-----------------|
| basic_info      | Object | 运行信息（MAC、时长、版本） |
| dev_info        | Object | CPU/内存/温度/风扇    |
| model           | String | 型号              |
| alias           | String | 别名              |
| sn              | String | 序列号             |
| kls_link_status | String | KiloLink 状态     |

12.1.2 设置设备别名

接口描述

按设备 MAC 修改拓扑中设备显示别名。

请求地址

/v1/switch/dev/alias/set

请求方法

POST

请求参数示例

```
1 {
```

```
2   "mac": "AA:BB:CC:DD:EE:FF",
3   "alias": "Camera-01"
4 }
```

请求参数说明

| 参数名称  | 参数类型   | 是否必填 | 描述     |
|-------|--------|------|--------|
| mac   | String | True | 设备 MAC |
| alias | String | True | 新别名    |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.2 端口管理

12.2.1 查询所有端口状态

接口描述

查询交换机全部端口状态（VLAN、Link、聚合、流量、EEE、PoE 等）。

请求地址

/v1/switch/port/dump

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
```

```

4  "data": [
5      {
6          "port": 11,
7          "alias": "port11",
8          "vlan": {
9              "type": "access",
10             "pvid": 3,
11             "vid": "3",
12             "alias": "vlan3",
13             "color": "#4298ED",
14             "config_file": ""
15         },
16         "link": {
17             "enable": true,
18             "link": true,
19             "media": "copper",
20             "speed_duplex": "1G",
21             "autonego": false,
22             "flow_control": true
23         },
24         "aggregation": {
25             "aggre_id": 0,
26             "type": "-"
27         },
28         "rate": {
29             "rx": {
30                 "packets": "5",
31                 "bytes": "1398",
32                 "error": "0",
33                 "loss": "0"
34             },
35             "tx": {
36                 "packets": "3",
37                 "bytes": "620",
38                 "error": "0",
39                 "loss": "0"
40             }
41         },
42         "eee": {
43             "actiphy": false,
44             "perfect_reach": false,
45             "enable": false
46         },
47         "poe": {
48             "enable": true,
49             "type": "PoE",
50             "power": "14.85",
51             "voltage": "54.02",
52             "current": "275"
53         },
54         "bind_ip": {
55             "enable": false,
56             "ip": ""
57         },
58         "igmp": {
59             "port": 11,
60             "router": false,
61             "uplink": true
62         },
63         "ptp_clock_domain": 0,

```

```

64         "device_ip": "192.168.3.38",
65         "model": "",
66         "image": "",
67         "company": ""
68     },
69     ...
70 ]
71 }

```

## 响应参数说明

data 为端口信息数组，每个元素对应一个端口的配置与状态。以下仅说明 data 内各字段（不含顶层的 code、message）。

### 说明约定

- data 中的 Object 字段**（如 vlan、link、rate 等）：先在本节**靠前位置**用独立表格列出其**子字段**；末尾「data 数组元素字段说明」表中对应行仅写「见 xxx」，不重复展开子字段。
- 非 Object 字段**（如 port、alias、ptp\_clock\_domain 等）：仅在「data 数组元素字段说明」表中描述。
- rate**：data[].rate 下的 rx、tx 均为对象且结构相同，因此使用两张表——「rate 对象字段说明」与「rate.rx / rate.tx 对象字段说明」。

### vlan 对象字段说明 (data[].vlan)

| 参数名称        | 参数类型   | 描述                                                                       |
|-------------|--------|--------------------------------------------------------------------------|
| type        | String | VLAN 类型：access 或 trunk                                                   |
| pvid        | Number | PVID / Native VLAN ID, 取值 1–4096                                         |
| vid         | String | access 时可为占位或与 PVID 相关字符串；trunk 时为允许通过的 Tagged VLAN 列表表达式（如 1,3,10-4000） |
| alias       | String | 该 VLAN 的别名                                                               |
| color       | String | 该 VLAN 的颜色（展示用字符串，如 #1D2024）                                             |
| config_file | String | 关联的配置文件标识；无则为空字符串                                                        |

### link 对象字段说明 (data[].link)

| 参数名称   | 参数类型    | 描述                        |
|--------|---------|---------------------------|
| enable | Boolean | 端口是否使能（逻辑开/关）             |
| link   | Boolean | 链路是否 Up（true 为在线）         |
| media  | String  | 介质类型：copper 电口 / fiber 光口 |

|              |         |                           |
|--------------|---------|---------------------------|
| speed_duplex | String  | 当前或协商速率双工，如 1G、10G、auto 等 |
| autonego     | Boolean | 是否启用自动协商                  |
| flow_control | Boolean | 是否开启流控                    |

### aggregation 对象字段说明 (data[].aggregation)

| 参数名称     | 参数类型   | 描述                                           |
|----------|--------|----------------------------------------------|
| aggre_id | Number | 所属聚合组 ID; 0 表示非聚合端口                          |
| type     | String | 聚合模式或展示占位 (如 static、lacp_active、- 等，以实际返回为准) |

### rate 对象字段说明 (data[].rate)

| 参数名称 | 参数类型   | 描述                                       |
|------|--------|------------------------------------------|
| rx   | Object | 接收方向流量统计，子字段见 [rate.rx / rate.tx 对象字段说明] |
| tx   | Object | 发送方向流量统计，子字段与 rx 相同                      |

### rate.rx / rate.tx 对象字段说明 (data[].rate.rx、data[].rate.tx)

| 参数名称    | 参数类型   | 描述                  |
|---------|--------|---------------------|
| packets | String | 报文个数 (示例中为十进制数字字符串) |
| bytes   | String | 字节数 (示例中为字符串)       |
| error   | String | 错包数                 |
| loss    | String | 丢包数                 |

### eee 对象字段说明 (data[].eee)

| 参数名称          | 参数类型    | 描述                  |
|---------------|---------|---------------------|
| actiphy       | Boolean | Actiphy 节能模式开关      |
| perfect_reach | Boolean | PerfectReach 节能模式开关 |
| enable        | Boolean | EEE 总开关 (与实现/展示一致)  |

### poe 对象字段说明 (data[].poe)

| 参数名称   | 参数类型    | 描述                 |
|--------|---------|--------------------|
| enable | Boolean | 是否启用 PoE 供电        |
| type   | String  | PoE 等级，如 PoE、PoE+、 |

|         |        |                 |
|---------|--------|-----------------|
|         |        | PoE++           |
| power   | String | 功率；未统计时可为空字符串   |
| voltage | String | 电压；未统计时可为空字符串   |
| current | String | 电流；可为 "0" 或空字符串 |

### bind\_ip 对象字段说明 (data[].bind\_ip)

| 参数名称   | 参数类型    | 描述                     |
|--------|---------|------------------------|
| enable | Boolean | 是否启用端口绑定静态 IP          |
| ip     | String  | 绑定的 IPv4 地址；未启用时可为空字符串 |

### igmp 对象字段说明 (data[].igmp)

| 参数名称   | 参数类型    | 描述                |
|--------|---------|-------------------|
| port   | Number  | 端口号（与条目端口一致）      |
| router | Boolean | IGMP 路由器相关角色/能力标识 |
| uplink | Boolean | 是否上行端口等拓扑角色标识     |

### data 数组元素字段说明

| 参数名称             | 参数类型   | 描述                                        |
|------------------|--------|-------------------------------------------|
| port             | Number | 端口 ID：1-24 为电口，33-36 为光口                  |
| alias            | String | 端口显示别名                                    |
| vlan             | Object | 见「vlan 对象字段说明」                            |
| link             | Object | 见「link 对象字段说明」                            |
| aggregation      | Object | 见「aggregation 对象字段说明」                     |
| rate             | Object | 见「rate 对象字段说明」及「rate.rx / rate.tx 对象字段说明」 |
| eee              | Object | 见「eee 对象字段说明」                             |
| poe              | Object | 见「poe 对象字段说明」                             |
| bind_ip          | Object | 见「bind_ip 对象字段说明」                         |
| igmp             | Object | 见「igmp 对象字段说明」                            |
| ptp_clock_domain | Number | PTP 时钟域，未使用时可为 0                          |
| device_ip        | String | 对端或关联设备 IP，无则为空字符串                        |
| model            | String | 关联设备型号，无则为空字符串                            |
| image            | String | 关联设备图标/镜像标识，无则为空字符串                       |

|              |        |                                 |
|--------------|--------|---------------------------------|
| company      | String | 关联设备厂商等信息，无则为空字符串               |
| connect_mode | String | （若返回）连接方式描述：普通端口或聚合组名称等；与实现扩展一致 |

### 12.2.2 端口控制

#### 接口描述

对端口执行 down / up / restart；不影响 PoE 供电策略本身。

#### 请求地址

/v1/switch/port/control

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "switch_id": 0,
3     "port_id": 1,
4     "command": "restart"
5 }
```

#### 请求参数说明

| 参数名称      | 参数类型   | 是否必填  | 描述                  |
|-----------|--------|-------|---------------------|
| switch_id | Number | False | 单台场景可填 0            |
| port_id   | Number | True  | 端口 ID               |
| command   | String | True  | down / up / restart |

#### 响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": ""
5 }
```

## 12.2.3 设置端口

### 接口描述

设置指定端口的别名、静态 IP、VLAN、Link、EEE、PoE 等。

### 请求地址

/v1/switch/port/set

### 请求方法

POST

### 请求参数示例

```
1 {
2   "msg": {
3     "alias": "port1",
4     "eee": {
5       "actiphy": false,
6       "perfect_reach": false,
7       "enable": true
8     },
9     "poe": {
10      "enable": true,
11      "type": "PoE",
12      "power": "",
13      "voltage": "",
14      "current": "0"
15    },
16    "link": {
17      "enable": true,
18      "link": true,
19      "media": "copper",
20      "speed_duplex": "1G",
21      "autonego": false,
22      "flow_control": true
23    },
24    "vlan": {
25      "type": "access",
26      "pvid": 1,
27      "vid": "1",
28      "alias": "vlan1",
29      "color": "#1D2024",
30      "config_file": ""
31    },
32    "ip": {},
33    "igmp": {
34      "port": 1,
35      "router": false,
36      "uplink": true
37    },
38    "port": 1
39  }
```

40 }

请求体为 {"msg": { ... }}, 下列参数均在 **msg** 对象内（与上例一致）。

请求参数说明

| 参数名称  | 参数类型   | 是否必填  | 描述                                                                                                              |
|-------|--------|-------|-----------------------------------------------------------------------------------------------------------------|
| port  | Number | True  | 待设置的端口号                                                                                                         |
| alias | String | False | 端口显示别名                                                                                                          |
| vlan  | Object | False | 请参考 12.2.1 查询所有端口信息<br>(switchPortDumpApi) 中的 <b>「vlan 对象字段说明」</b> 。                                            |
| link  | Object | False | 请参考 12.2.1 中的 <b>「link 对象字段说明」</b> 。                                                                            |
| eee   | Object | False | 请参考 12.2.1 中的 <b>「eee 对象字段说明」</b> 。                                                                             |
| poe   | Object | False | 请参考 12.2.1 中的 <b>「poe 对象字段说明」</b> 。                                                                             |
| ip    | Object | False | 静态 IP / 绑定；子字段含 enable、addr、op（绑定组）。与查询侧 <b>「bind_ip 对象字段说明」</b> （12.2.1）对照时请注意：查询为 bind_ip，本接口请求字段名为 ip，且多 op。 |
| igmp  | Object | False | 请参考 12.2.1 中的 <b>「igmp 对象字段说明」</b> 。                                                                            |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.2.4 批量设置端口

接口描述

对多个端口批量设置使能、速率、EEE、PoE 等。

请求地址

/v1/switch/port/set/batch

请求方法

POST

请求参数示例

```
1 {
2   "msg": {
3     "eee_enable": true,
4     "poe_enable": true,
5     "port_enable": true,
6     "speed_duplex": "auto",
7     "uplink": false,
8     "vlan": {
9       "alias": "",
10      "color": "",
11      "config_name": "",
12      "pvid": 1,
13      "type": "access",
14      "vid": "",
15      "interface": {}
16    },
17    "ports": [
18      1,
19      3
20    ]
21  }
22 }
```

请求参数说明

请求体为 {"msg":{ ... }}, 下列参数均在 **msg** 内。

msg.vlan 对象字段说明

| 参数名称        | 参数类型   | 是否必填  | 描述                                                                                                                                             |
|-------------|--------|-------|------------------------------------------------------------------------------------------------------------------------------------------------|
| (结构与含义)     | -      | -     | type、pvid、vid、alias、color 等请参考 12.2.1 查询所有端口信息 (switchPortDumpApi) 中的 <b>「vlan 对象字段说明」</b> ；批量示例中的 config_name 等与查询侧 config_file 命名可能不同，以网关为准。 |
| config_name | String | False | 配置名称（示例字段）                                                                                                                                     |
| interface   | Object | False | VLAN 三层接口；子字段                                                                                                                                  |

|  |  |  |                                                                                   |
|--|--|--|-----------------------------------------------------------------------------------|
|  |  |  | 请参考 <b>12.4.1</b> 获取所有 VLAN<br>(switchVlanDumpApi) 中的 <b>「interface 对象字段说明」</b> 。 |
|--|--|--|-----------------------------------------------------------------------------------|

msg 根级字段说明

| 参数名称         | 参数类型     | 是否必填  | 描述                       |
|--------------|----------|-------|--------------------------|
| ports        | Number[] | True  | 待批量设置的端口 ID 列表           |
| port_enable  | Boolean  | False | 端口使能                     |
| speed_duplex | String   | False | auto、10G、1G、100MF、10MF 等 |
| eee_enable   | Boolean  | False | EEE 开关                   |
| poe_enable   | Boolean  | False | PoE 开关                   |
| uplink       | Boolean  | False | 是否上行端口等(示例字段，以网关实现为准)    |
| vlan         | Object   | False | 见上「msg.vlan 对象字段说明」      |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.2.5 设置端口别名

接口描述

仅修改端口别名。

请求地址

/v1/switch/port/alias/set

请求方法

POST

请求参数示例

```
1 {
2   "port": 1,
3   "alias": "port-1"
4 }
```

请求参数说明

| 参数名称  | 参数类型   | 是否必填 | 描述  |
|-------|--------|------|-----|
| port  | Number | True | 端口号 |
| alias | String | True | 别名  |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.2.6 查询端口流量

接口描述

查询指定端口的收发流量统计。

请求地址

/v1/switch/port/traffic

请求方法

POST

请求参数示例

```
1 {
2   "port": 1
3 }
```

请求参数说明

| 参数名称 | 参数类型   | 是否必填 | 描述  |
|------|--------|------|-----|
| port | Number | True | 端口号 |

响应示例

```
1 {
```

```

2  "code": 200,
3  "message": "ok",
4  "data": {
5      "port": "1",
6      "traffic": [
7          {
8              "rx": {
9                  "packets": "8438",
10                 "bytes": "12236088",
11                 "error": "0",
12                 "loss": "0"
13             },
14             "tx": {
15                 "packets": "29",
16                 "bytes": "10757",
17                 "error": "0",
18                 "loss": "0"
19             }
20         },
21         ...
22     ]
23 }
24 }

```

## 响应参数说明

### data.traffic[] 元素字段说明 (data.traffic 数组中每一项)

| 参数名称 | 参数类型   | 描述                                                                                 |
|------|--------|------------------------------------------------------------------------------------|
| rx   | Object | 接收统计；子字段请参考 <b>12.2.1 「rate.rx / rate.tx 对象字段说明」</b> (packets、bytes、error、loss 等)。 |
| tx   | Object | 发送统计；子字段同上。                                                                        |

### data 对象字段说明

| 参数名称    | 参数类型   | 描述                                          |
|---------|--------|---------------------------------------------|
| port    | String | 查询的端口号 (示例为字符串)                             |
| traffic | Array  | 流量统计序列 (如多时段/多组)；元素见「data.traffic[] 元素字段说明」 |
| 参数名称    | 参数类型   | 描述                                          |
| code    | Number | 业务状态码                                       |
| message | String | 提示信息                                        |
| data    | Object | 见上「data 对象字段说明」                             |

## 12.2.7 查询端口流量 top

### 接口描述

查询所有端口的流量统计列表。

### 请求地址

/v1/switch/port/traffic/query/top

### 请求方法

GET

### 请求参数示例

无

### 请求参数说明

无

### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "sum_rx_rate": 93.395,
6     "sum_tx_rate": 0.24,
7     "desc_stat": [
8       {
9         "port": "1",
10        "rx_rate": 93.318,
11        "tx_rate": 0.036,
12        "top_rate": 93.318,
13        "top_usage": 0.093
14      },
15      {
16        "port": "23",
17        "rx_rate": 0.032,
18        "tx_rate": 0.09,
19        "top_rate": 0.09,
20        "top_usage": 0
21      },
22      {
23        "port": "4",
24        "rx_rate": 0.019,
25        "tx_rate": 0.06,
26        "top_rate": 0.06,
27        "top_usage": 0
28      },
29    ]
30  }
```

```

29     {
30         "port": "33",
31         "rx_rate": 0.003,
32         "tx_rate": 0.045,
33         "top_rate": 0.045,
34         "top_usage": 0
35     },
36     {
37         "port": "11",
38         "rx_rate": 0.016,
39         "tx_rate": 0.005,
40         "top_rate": 0.016,
41         "top_usage": 0
42     },
43     {
44         "port": "13",
45         "rx_rate": 0.007,
46         "tx_rate": 0.004,
47         "top_rate": 0.007,
48         "top_usage": 0
49     }
50 ]
51 }
52 }

```

响应参数说明

data.desc\_stat[] 元素字段说明

| 参数名称      | 参数类型   | 描述                     |
|-----------|--------|------------------------|
| port      | String | 端口号                    |
| rx_rate   | Number | 接收速率（示例单位与网关一致，如 Mbps） |
| tx_rate   | Number | 发送速率                   |
| top_rate  | Number | 用于排序的峰值/综合速率           |
| top_usage | Number | 占用或利用率相关指标             |

data 对象字段说明

| 参数名称        | 参数类型   | 描述                                    |
|-------------|--------|---------------------------------------|
| sum_rx_rate | Number | 各端口接收速率汇总                             |
| sum_tx_rate | Number | 各端口发送速率汇总                             |
| desc_stat   | Array  | Top 端口列表，元素见「data.desc_stat[] 元素字段说明」 |
| 参数名称        | 参数类型   | 描述                                    |
| code        | Number | 业务状态码                                 |
| message     | String | 提示信息                                  |
| data        | Object | 见上「data 对象字段说明」                       |

# 12.2.8 设置关注端口

## 接口描述

设置关注的端口列表

## 请求地址

/v1/switch/port/display/set

## 请求方法

POST

## 请求参数示例

```
1 {
2     "type": "select",
3     "ports": [
4         1,
5         11,
6         13,
7         23,
8         33,
9         4
10    ]
11 }
```

## 请求参数说明

| 参数名称  | 参数类型     | 是否必填 | 描述                       |
|-------|----------|------|--------------------------|
| type  | String   | True | 关注类型；示例为 select, 以网关约定为准 |
| ports | Number[] | True | 要设为关注的端口 ID 列表           |

## 响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": null
5 }
```

## 响应参数说明

| 参数名称 | 参数类型   | 描述             |
|------|--------|----------------|
| code | Number | 业务状态码，200 表示成功 |

|         |        |            |
|---------|--------|------------|
| message | String | 提示信息       |
| data    | Null   | 成功时可为 null |

## 12.2.9 删除关注端口

### 接口描述

从关注的端口列表中删除端口

### 请求地址

/v1/switch/port/display/del

### 请求方法

POST

### 请求参数示例

```
1 {
2     "type": "select",
3     "ports": [
4         4
5     ]
6 }
```

### 请求参数说明

| 参数名称  | 参数类型     | 是否必填 | 描述                      |
|-------|----------|------|-------------------------|
| type  | String   | True | 关注类型；示例为 select，须与设置时一致 |
| ports | Number[] | True | 要从关注列表移除的端口 ID 列表       |

### 响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": null
5 }
```

### 响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | Number | 业务状态码，200 表示成功 |
| message | String | 提示信息           |

|      |      |            |
|------|------|------------|
| data | Null | 成功时可为 null |
|------|------|------------|

### 12.2.10 查询关注端口信息

#### 接口描述

查询关注端口的状态

#### 请求地址

/v1/switch/port/display/query

#### 请求方法

GET

#### 请求参数示例

无

#### 请求参数说明

无

#### 响应示例

```
1 {
2   "cascade_ports": [],
3   "select_ports": [
4     {
5       "port": 11,
6       "alias": "port11",
7       "vlan": {
8         "type": "access",
9         "pvid": 3,
10        "vid": "3",
11        "alias": "vlan3",
12        "color": "#4298ED",
13        "config_file": ""
14      },
15      "link": {
16        "enable": true,
17        "link": true,
18        "media": "copper",
19        "speed_duplex": "1G",
20        "autonego": false,
21        "flow_control": true
22      },
23      "aggregation": {
24        "aggre_id": 0,
25        "type": "-"
26      }
27    }
28  ]
29 }
```

```
26     },
27     "rate": {
28         "rx": {
29             "packets": "6",
30             "bytes": "1789",
31             "error": "0",
32             "loss": "0"
33         },
34         "tx": {
35             "packets": "4",
36             "bytes": "800",
37             "error": "0",
38             "loss": "0"
39         }
40     },
41     "eee": {
42         "actiphy": false,
43         "perfect_reach": false,
44         "enable": false
45     },
46     "poe": {
47         "enable": true,
48         "type": "PoE",
49         "power": "14.48",
50         "voltage": "54.02",
51         "current": "268"
52     },
53     "bind_ip": {
54         "enable": false,
55         "ip": ""
56     },
57     "igmp": {
58         "port": 11,
59         "router": false,
60         "uplink": true
61     },
62     "ptp_clock_domain": 0,
63     "device_ip": "192.168.3.38",
64     "model": "N60",
65     "image": "./static/img/N60.png",
66     "company": "Kiloview"
67 },
68 ...
69 ]
70 }
```

响应参数说明

示例为 **cascade\_ports / select\_ports 业务负载**；若网关返回标准信封 { "code", "message", "data" }, 则下列两字段位于 **data** 内，并可在表末补充顶层 code、message。

**select\_ports[]、cascade\_ports[] 元素字段说明**（两数组元素结构相同，与 **12.2.1** 查询所有端口信息 (switchPortDumpApi) 中 **data[] 单条端口**一致)

| 参数名称 | 参数类型   | 描述    |
|------|--------|-------|
| port | Number | 端口 ID |

|                  |        |                                                          |
|------------------|--------|----------------------------------------------------------|
| alias            | String | 端口别名                                                     |
| vlan             | Object | 请参考 12.2.1 查询所有端口信息 (switchPortDumpApi) 中的「vlan 对象字段说明」。 |
| link             | Object | 请参考 12.2.1 中的「link 对象字段说明」。                              |
| aggregation      | Object | 请参考 12.2.1 中的「aggregation 对象字段说明」。                       |
| rate             | Object | 请参考 12.2.1 中的「rate 对象字段说明」及「rate.rx / rate.tx 对象字段说明」。   |
| eee              | Object | 请参考 12.2.1 中的「eee 对象字段说明」。                               |
| poe              | Object | 请参考 12.2.1 中的「poe 对象字段说明」。                               |
| bind_ip          | Object | 请参考 12.2.1 中的「bind_ip 对象字段说明」。                           |
| igmp             | Object | 请参考 12.2.1 中的「igmp 对象字段说明」。                              |
| ptp_clock_domain | Number | PTP 时钟域等                                                 |
| device_ip        | String | 对端/设备 IP                                                 |
| model            | String | 对端设备型号                                                   |
| image            | String | 图标或资源路径                                                  |
| company          | String | 厂商等展示字段                                                  |
| connect_mode     | String | (若返回) 连接方式说明                                             |

#### data (或响应根级) 对象字段说明

| 参数名称          | 参数类型   | 描述                                                       |
|---------------|--------|----------------------------------------------------------|
| cascade_ports | Array  | 级联/关联展示用端口列表；元素见上「select_ports[]、cascade_ports[] 元素字段说明」 |
| select_ports  | Array  | 用户选中关注端口列表；元素结构同上                                        |
| 参数名称          | 参数类型   | 描述                                                       |
| code          | Number | (若返回标准信封) 业务状态码                                          |
| message       | String | (若返回标准信封) 提示信息                                           |
| data          | Object | (若返回标准信封) 内含 cascade_ports、select_ports                  |

# 12.2.11 移除设备

## 接口描述

移除某交换机端口上所连设备（进入移除流程，可配置生效时长）。

## 请求地址

/v1/switch/device/remove

## 请求方法

POST

## 请求参数示例

```
1 {
2   "switch_sn": "",
3   "port": 1,
4   "duration": 10
5 }
```

## 请求参数说明

| 参数名称      | 参数类型   | 是否必填  | 描述                         |
|-----------|--------|-------|----------------------------|
| switch_sn | String | False | 序列号;单机可传空字符串表示本机           |
| port      | Number | True  | 端口 ID                      |
| duration  | Number | False | 生效时间 (分钟) ; 0 表示使用默认 10 分钟 |

## 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

# 12.3 端口聚合

## 12.3.1 创建链路聚合组

### 接口描述

创建聚合组（单次请求仅支持创建一个聚合组）

补充说明：

- aggre\_id 在设备内需唯一，建议使用连续编号（如 1、2、3...）；不传或传 0 时由后端自动生成
- type 取值：static（静态）、lacp\_active（LACP 主动）
- members 为待加入该聚合组的端口 ID 列表

请求地址

/v1/switch/aggr/add

请求方法

POST

请求参数示例

```
1 {
2     "msg": [
3         {
4             "aggre_id": 1,
5             "alias": "上联聚合组",
6             "type": "lacp_active",
7             "members": [1, 2]
8         }
9     ]
10 }
```

请求参数说明

| 参数名称         | 参数类型   | 是否必填  | 描述                                   |
|--------------|--------|-------|--------------------------------------|
| msg          | array  | True  | 聚合组配置列表(单次请求仅支持一个聚合组)                |
| msg.aggre_id | int32  | True  | 聚合组 ID（传入 0 会自动计算 id)                |
| msg.alias    | string | False | 聚合组别名                                |
| msg.type     | string | True  | 聚合类型:static(静态)、lacp_active（LACP 主动) |
| msg.members  | array  | True  | 要加入聚合组的端口 ID 列表                      |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok"
4 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

12.3.2 删除链路聚合组

接口描述

删除指定聚合组（可批量删除）

请求地址

/v1/switch/aggr/del

请求方法

POST

请求参数示例

```
1 {
2   "aggre_id": [1, 2]
3 }
```

请求参数说明

| 参数名称     | 参数类型  | 是否必填 | 描述            |
|----------|-------|------|---------------|
| aggre_id | array | True | 要删除的聚合组 ID 列表 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok"
4 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

### 12.3.3 查询全部链路聚合组

#### 接口描述

查询所有聚合组信息

补充说明：

- configured：配置层面已加入该聚合组的端口列表
- aggregated：运行状态下实际聚合成功的端口列表（可能少于 configured）

#### 请求地址

/v1/switch/aggr/dump

#### 请求方法

GET

#### 请求参数示例

```
1 {}
```

#### 请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "aggre_id": 1,
7       "alias": "聚合组 1",
8       "type": "static",
9       "configured": [1, 2, 3],
10      "aggregated": [1, 2, 3]
11    }
12  ]
13 }
```

#### 响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

|                 |        |                          |
|-----------------|--------|--------------------------|
| data            | array  | 聚合组列表                    |
| data.aggre_id   | int32  | 聚合组 ID                   |
| data.alias      | string | 聚合组别名                    |
| data.type       | string | 聚合类型: static/lacp_active |
| data.configured | array  | 已配置加入聚合组的端口 ID 列表        |
| data.aggregated | array  | 实际成功加入聚合组的端口 ID 列表       |

### 12.3.4 查询指定或全部链路聚合组

#### 接口描述

按聚合组 ID 查询指定聚合组信息（支持传入多个 ID 批量查询）；当不传 aggre\_id 或为空数组时返回空列表。如需查询全部聚合组请使用 /v1/switch/aggr/dump 接口。

#### 请求地址

/v1/switch/aggr/get

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "aggre_id": [1]
3 }
```

#### 请求参数说明

| 参数名称     | 参数类型  | 是否必填  | 描述                                                      |
|----------|-------|-------|---------------------------------------------------------|
| aggre_id | array | False | 要查询的聚合组 ID 列表；<br>为空时返回空列表，查询全部请使用 /v1/switch/aggr/dump |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "aggre_id": 1,
7       "alias": "string",
```

```

8         "type": "string",
9         "configured": [
10            1
11        ],
12        "aggregated": [
13            1
14        ]
15    }
16 ]
17 }

```

## 响应参数说明

| 参数名称            | 参数类型   | 描述                      |
|-----------------|--------|-------------------------|
| code            | int32  | 业务状态码，200 表示成功          |
| message         | string | 业务提示信息                  |
| data            | array  | 业务数据主体                  |
| data.aggre_id   | int32  | 数值参数                    |
| data.alias      | string | 别名                      |
| data.type       | string | 聚合类型：static/lacp_active |
| data.configured | array  | 已配置加入聚合组的端口 ID 列表       |
| data.aggregated | array  | 实际成功加入聚合组的端口 ID 列表      |

## 12.3.5 向聚合组追加端口

### 接口描述

向指定聚合组追加端口成员（不改变聚合组类型）

### 请求地址

/v1/switch/aggr/port/add

### 请求方法

POST

### 请求参数示例

```

1 {
2     "aggre_id": 1,
3     "members": [3, 4]
4 }

```

请求参数说明

| 参数名称     | 参数类型  | 是否必填 | 描述               |
|----------|-------|------|------------------|
| aggre_id | int32 | True | 目标聚合组 ID         |
| members  | array | True | 要追加到聚合组的端口 ID 列表 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {}
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

12.3.6 修改链路聚合组配置

接口描述

修改聚合组配置（支持修改别名、类型、成员端口，单次请求仅支持修改一个聚合组）

请求地址

/v1/switch/aggr/set

请求方法

POST

请求参数示例

```
1 {
2     "msg": [
3         {
4             "aggre_id": 1,
5             "alias": "上联聚合组 A",
6             "type": "lacp_passive",
7             "members": [1, 2, 3]
8         }
9     ]
10 }
```

请求参数说明

| 参数名称         | 参数类型   | 是否必填  | 描述                          |
|--------------|--------|-------|-----------------------------|
| msg          | array  | True  | 聚合组配置列表(单次请求仅支持一个聚合组)       |
| msg.aggre_id | int32  | True  | 需要修改的聚合组 ID                 |
| msg.alias    | string | False | 聚合组别名                       |
| msg.type     | string | False | 聚合类型：<br>static/lacp_active |
| msg.members  | array  | False | 更新后的成员端口 ID 列表              |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {}
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

12.4 VLAN 管理

12.4.1 查询所有 VLAN 信息

接口描述

获取当前所有 VLAN 属性及端口、接口等信息。

请求地址

/v1/switch/vlan/dump

请求方法

GET

## 请求参数示例

无

## 请求参数说明

无

## 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "id": 1,
7       "alias": "vlan1",
8       "color": "#1D2024",
9       "configs": [],
10      "config_names": [],
11      "ports": [
12        1,
13        2,
14        3,
15        4,
16        5,
17        6,
18        33,
19        34,
20        35,
21        36
22      ],
23      "interface": {
24        "enable": true,
25        "ip": "172.30.46.107",
26        "netmask": "255.255.255.0",
27        "type": "dhcp",
28        "referenced": true,
29        "is_default_route": false
30      },
31      "dhcp_enable": false,
32      "vlan_if": {
33        "vlan_id": 1,
34        "enable": true,
35        "ip": "172.30.46.254",
36        "netmask": "255.255.255.0"
37      },
38      "dhcp_conflict": false
39    },
40    {
41      "id": 3,
42      "alias": "vlan3",
43      "color": "#4298ED",
44      "configs": [],
45      "config_names": [],
46      "ports": [
47        7,
```

```

48         8,
49         9,
50         10,
51         11,
52         12,
53         13,
54         14,
55         15,
56         16,
57         17,
58         18,
59         19,
60         20,
61         21,
62         22,
63         23,
64         24
65     ],
66     "interface": {
67         "enable": true,
68         "ip": "192.168.3.2",
69         "netmask": "255.255.255.0",
70         "type": "dhcp",
71         "referenced": true,
72         "is_default_route": false
73     },
74     "dhcp_enable": true,
75     "vlan_if": {
76         "vlan_id": 3,
77         "enable": true,
78         "ip": "192.168.3.1",
79         "netmask": "255.255.255.0"
80     },
81     "dhcp_conflict": false
82 }
83 ]
84 }

```

## 响应参数说明

Object 类型字段先列子表，再在「data 数组元素」总表中引用。

**[interface 对象字段说明]**（data[].interface，VLAN 三层/管理接口展示）

| 参数名称             | 参数类型    | 描述                 |
|------------------|---------|--------------------|
| enable           | Boolean | 是否启用该接口配置          |
| ip               | String  | IPv4 地址；未配置时可为空字符串 |
| netmask          | String  | 子网掩码               |
| type             | String  | 地址类型，如 dhcp、manual |
| referenced       | Boolean | 是否被引用/关联（业务展示字段）   |
| is_default_route | Boolean | 是否为默认路由相关接口        |

### 「vlan\_if 对象字段说明」 (data[].vlan\_if, VLANIF 下发/运行参数)

| 参数名称    | 参数类型    | 描述          |
|---------|---------|-------------|
| vlan_id | Number  | 对应 VLAN ID  |
| enable  | Boolean | VLANIF 是否启用 |
| ip      | String  | VLANIF IP   |
| netmask | String  | VLANIF 掩码   |

### 「data 数组元素字段说明」 (data[] 中每一条 VLAN)

| 参数名称          | 参数类型     | 描述                                                                        |
|---------------|----------|---------------------------------------------------------------------------|
| id            | Number   | VLAN ID, 取值 1–4096                                                        |
| alias         | String   | VLAN 别名                                                                   |
| color         | String   | 展示用颜色字符串                                                                  |
| configs       | Number[] | 关联配置文件 ID 列表                                                              |
| config_names  | String[] | 关联配置文件名称列表 (与 configs 对应, 以网关为准)                                          |
| ports         | Number[] | 加入该 VLAN 的端口 ID 列表                                                        |
| interface     | Object   | 见 「interface 对象字段说明」                                                      |
| dhcp_enable   | Boolean  | 是否启用 DHCP (示例字段)                                                          |
| vlan_if       | Object   | 见 「vlan_if 对象字段说明」                                                        |
| dhcp_conflict | Boolean  | DHCP 冲突等业务标志 (示例字段)                                                       |
| 参数名称          | 参数类型     | 描述                                                                        |
| code          | Number   | 业务状态码, 200 表示成功                                                           |
| message       | String   | 提示信息                                                                      |
| data          | Array    | VLAN 列表; 元素见 「data 数组元素字段说明」 (与 switch_api.go 中 switchVlanAttr 及网关扩展字段一致) |

## 12.4.2 添加 VLAN

### 接口描述

添加一个或多个 VLAN; 若 VLAN 已存在则覆盖。

### 请求地址

/v1/switch/vlan/add

请求方法

POST

请求参数示例

```
1 {
2   "data": [
3     {
4       "id": 10,
5       "configs": [
6         1
7       ],
8       "color": "#38bb70",
9       "alias": "vlan10",
10
11      "ports": [ ],
12
13      "vlan_if": {
14        "vlan_id":1,
15        "ip": "",
16        "enable": false,
17        "netmask": "255.255.255.0"
18      },
19      "interface": {
20        "enable": false,
21        "ip": "",
22        "netmask": "255.255.255.0",
23        "type": "dhcp"
24      }
25    }
26  ]
27 }
```

请求参数说明

| 参数名称 | 参数类型  | 是否必填 | 描述                                                                                                                                                      |
|------|-------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| data | Array | True | VLAN 对象数组； <b>每个元素的字段与 12.2.1 查询所有VLAN信息 中的「data 数组元素字段说明」一致</b> ；其中 interface、vlan_if 子字段分别见同节的「interface 对象字段说明」、「vlan_if 对象字段说明」。未在请求中出现的字段由网关按默认处理。 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": ""
5 }
```

错误时 code 可能为 70001 等，message 含中英文说明。

12.4.3 设置 VLAN

接口描述

编辑指定 VLAN 的配置。

请求地址

/v1/switch/vlan/set

请求方法

POST

请求参数示例

```
1 {
2     "data": {
3         "id": 10,
4         "configs": [
5             1
6         ],
7         "color": "#FFD700",
8         "alias": "vlan10",
9         "ports": [],
10        "vlan_if": {
11            "vlan_id": 10,
12            "enable": false,
13            "ip": "",
14            "netmask": ""
15        }
16    }
17 }
```

请求参数说明

| 参数名称 | 参数类型   | 是否必填 | 描述                                                    |
|------|--------|------|-------------------------------------------------------|
| data | Object | True | 单条 VLAN；字段与 12.2.1 查询所有 VLAN 信息中的 「data 数组元素字段说明」 一致。 |

|  |  |  |                                                                                 |
|--|--|--|---------------------------------------------------------------------------------|
|  |  |  | interface、vlan_if 见 12.2.1 查询所有 VLAN 信息中的 「interface 对象字段说明」、 「vlan_if 对象字段说明」。 |
|--|--|--|---------------------------------------------------------------------------------|

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": ""
5 }
```

12.4.4 查询 VLAN 信息

接口描述

按 VLAN ID 查询单条 VLAN 信息。

请求地址

/v1/switch/vlan/query

请求方法

POST

请求参数示例

```
1 {
2     "vlan_id": 100
3 }
```

请求参数说明

| 参数名称    | 参数类型   | 是否必填 | 描述      |
|---------|--------|------|---------|
| vlan_id | Number | True | VLAN ID |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {
5         "id": 10,
```

```
6     "alias": "vlan10",
7     "color": "#38bb70",
8     "configs": [
9         1
10    ],
11    "ports": [33],
12    "vlan_if": {
13        "vlan_id": 10,
14        "enable": false,
15        "ip": "",
16        "netmask": ""
17    }
18 }
19 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述                                                                                                                 |
|---------|--------|--------------------------------------------------------------------------------------------------------------------|
| code    | Number | 业务状态码                                                                                                              |
| message | String | 提示信息                                                                                                               |
| data    | Object | 单条 VLAN；字段与 12.2.1 查询所有 VLAN 信息中的「data 数组元素字段说明」一致（查询返回可能省略部分扩展字段）。interface、vlan_if 见 12.2.1 查询所有 VLAN 信息中的 对应子表。 |

12.4.5 删除 VLAN

接口描述

删除一个或多个 VLAN。

请求地址

/v1/switch/vlan/del

请求方法

POST

请求参数示例

```
1 {
2     "vlans": [
3         10
4     ]
5 }
```

请求参数说明

| 参数名称   | 参数类型     | 是否必填 | 描述              |
|--------|----------|------|-----------------|
| vlangs | Number[] | True | 待删除的 VLAN ID 列表 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | Number | 业务状态码          |
| message | String | 提示信息           |
| data    | Null   | 成功时业务数据可为 null |

12.4.6 设置 VLAN 端口

接口描述

将指定端口批量加入对应 VLAN。

请求地址

/v1/switch/vlan/port/set

请求方法

POST

请求参数示例

```
1 {
2   "msg": [
3     {
4       "vlan_id": 10,
5       "ports": [
6         36,
7         35
8       ]
9     }
10  ]
11 }
```

请求参数说明

请求体为 **msg** 数组；每一项表示将若干端口加入指定 VLAN。

**msg[] 元素字段说明**

| 参数名称    | 参数类型     | 是否必填 | 描述                  |
|---------|----------|------|---------------------|
| vlan_id | Number   | True | VLAN ID             |
| ports   | Number[] | True | 要加入该 VLAN 的端口 ID 列表 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": ""
5 }
```

12.5 DHCP

12.5.1 查询 DHCP 客户端列表

接口描述

查询 DHCP 客户端列表（获取 IP 的设备）

请求地址

/v1/switch/dhcps/clients/dump

请求方法

POST

请求参数示例

```
1 {
2     "vlan_id": [ ]
3 }
```

请求参数说明

| 参数名称    | 参数类型  | 是否必填  | 描述                          |
|---------|-------|-------|-----------------------------|
| vlan_id | array | False | VLAN ID 列表，不传则查询所有 VLAN 的客户 |

|  |  |  |   |
|--|--|--|---|
|  |  |  | 端 |
|--|--|--|---|

## 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "vlan_id": 1,
7       "ip": "string",
8       "netmask": "string",
9       "type": "string",
10      "host_name": "string",
11      "mac": "string",
12      "expired_time": {
13        "days": 1,
14        "hours": 1,
15        "minutes": 1
16      },
17      "brand": "string",
18      "mode": "string",
19      "sn": "string",
20      "connect_port": 1,
21      "bind_type": 1,
22      "color": "string"
23    }
24  ]
25 }
```

## 响应参数说明

| 参数名称                      | 参数类型   | 描述                                    |
|---------------------------|--------|---------------------------------------|
| code                      | int32  | 业务状态码，200 表示成功                        |
| message                   | string | 业务提示信息                                |
| data                      | array  | DHCP 客户端列表                            |
| data.vlan_id              | int32  | 客户端所属 VLAN ID                         |
| data.ip                   | string | 分配的 IP 地址                             |
| data.netmask              | string | 子网掩码                                  |
| data.type                 | string | IP 分配类型：Automatic（自动分配）或 Manual（手动绑定） |
| data.host_name            | string | 客户端主机名                                |
| data.mac                  | string | 客户端 MAC 地址                            |
| data.expired_time         | object | 租约剩余时间                                |
| data.expired_time.days    | int32  | 剩余天数                                  |
| data.expired_time.hours   | int32  | 剩余小时数                                 |
| data.expired_time.minutes | int32  | 剩余分钟数                                 |
| data.brand                | string | 设备品牌/厂商                               |
| data.mode                 | string | 设备型号                                  |

|                   |        |                                             |
|-------------------|--------|---------------------------------------------|
| data.sn           | string | 设备序列号                                       |
| data.connect_port | int32  | 连接的交换机端口号                                   |
| data.bind_type    | int32  | 绑定类型: 0-未绑定, 1-IP 绑定, 2-MAC 绑定, 3-IP+MAC 绑定 |
| data.color        | string | VLAN 颜色                                     |

- GET /v1/switch/dhcps/get - 获取 DHCP 配置
- POST /v1/switch/dhcps/set - 设置 DHCP

### 12.5.2 获取指定 VLAN 的 VLAN 接口与 DHCP 配置

#### 接口描述

获取指定 VLAN 的 VLAN 接口和 DHCP 服务器配置

#### 请求地址

/v1/switch/dhcps/get

#### 请求方法

GET

#### 请求参数示例

```
1 /v1/switch/dhcps/get?vlan_id=10
```

#### 请求参数说明

| 参数名称    | 参数类型  | 是否必填 | 描述                  |
|---------|-------|------|---------------------|
| vlan_id | int32 | True | VLAN ID (位置: query) |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "vlan_if": {
6       "vlan_id": 1,
7       "enable": true,
8       "ip": "192.168.1.1",
9       "netmask": "255.255.255.0"
10    },
11    "dhcp": {
12      "vlan_id": 1,
13      "enable": true,
```

```

14         "ip_pool": {
15             "start": "192.168.1.100",
16             "end": "192.168.1.200"
17         },
18         "gateway": {
19             "type": "default",
20             "ip": "192.168.1.1"
21         },
22         "dns": "8.8.8.8",
23         "lease_time": {
24             "days": 1,
25             "hours": 0,
26             "minutes": 0
27         }
28     }
29 }
30 }

```

## 响应参数说明

| 参数名称                         | 参数类型   | 描述                                     |
|------------------------------|--------|----------------------------------------|
| code                         | int32  | 业务状态码，200 表示成功                         |
| message                      | string | 业务提示信息                                 |
| data                         | object | VLAN 接口和 DHCP 配置信息                     |
| data.vlan_if                 | object | VLAN 接口配置                              |
| data.vlan_if.vlan_id         | int32  | VLAN ID                                |
| data.vlan_if.enable          | bool   | VLAN 接口使能开关                            |
| data.vlan_if.ip              | string | VLAN 接口 IP 地址                          |
| data.vlan_if.netmask         | string | VLAN 接口子网掩码                            |
| data.dhcp                    | object | DHCP 服务器配置                             |
| data.dhcp.vlan_id            | int32  | VLAN ID                                |
| data.dhcp.enable             | bool   | DHCP 服务器使能开关                           |
| data.dhcp.ip_pool            | object | DHCP 地址池                               |
| data.dhcp.ip_pool.start      | string | 地址池起始 IP                               |
| data.dhcp.ip_pool.end        | string | 地址池结束 IP                               |
| data.dhcp.gateway            | object | 网关配置                                   |
| data.dhcp.gateway.type       | string | 网关类型：default（使用 VLAN IP）或 manual（手动指定） |
| data.dhcp.gateway.ip         | string | 网关 IP 地址                               |
| data.dhcp.dns                | string | DNS 服务器地址                              |
| data.dhcp.lease_time         | object | 租约时间                                   |
| data.dhcp.lease_time.days    | int32  | 租约天数                                   |
| data.dhcp.lease_time.hours   | int32  | 租约小时数                                  |
| data.dhcp.lease_time.minutes | int32  | 租约分钟数                                  |

### 12.5.3 设置指定 VLAN 的 VLAN 接口与 DHCP 配置

#### 接口描述

设置指定 VLAN 的 VLAN 接口和 DHCP 服务器配置

#### 请求地址

/v1/switch/dhcps/set

#### 请求方法

POST

#### 请求参数示例

```
1 {}
```

#### 请求参数说明

| 参数名称                    | 参数类型   | 是否必填  | 描述                          |
|-------------------------|--------|-------|-----------------------------|
| vlan_if                 | object | True  | VLAN 接口配置                   |
| vlan_if.vlan_id         | int32  | True  | VLAN ID                     |
| vlan_if.enable          | bool   | True  | VLAN 接口使能开关                 |
| vlan_if.ip              | string | True  | VLAN 接口 IP 地址               |
| vlan_if.netmask         | string | True  | VLAN 接口子网掩码                 |
| dhcp                    | object | True  | DHCP 服务器配置                  |
| dhcp.vlan_id            | int32  | True  | VLAN ID                     |
| dhcp.enable             | bool   | True  | DHCP 服务器使能开关                |
| dhcp.ip_pool            | object | True  | DHCP 地址池                    |
| dhcp.ip_pool.start      | string | True  | 地址池起始 IP                    |
| dhcp.ip_pool.end        | string | True  | 地址池结束 IP                    |
| dhcp.gateway            | object | True  | 网关配置                        |
| dhcp.gateway.type       | string | True  | 网关类型：default 或 manual       |
| dhcp.gateway.ip         | string | True  | 网关 IP 地址（type 为 manual 时必填） |
| dhcp.dns                | string | False | DNS 服务器地址                   |
| dhcp.lease_time         | object | True  | 租约时间                        |
| dhcp.lease_time.days    | int32  | True  | 租约天数                        |
| dhcp.lease_time.hours   | int32  | True  | 租约小时数                       |
| dhcp.lease_time.minutes | int32  | True  | 租约分钟数                       |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok"
4 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

- POST /v1/switch/dhcps/enable/set - 启用/禁用 DHCP

12.5.4 开启或关闭指定 VLAN 的 DHCP 服务

接口描述

开启/关闭指定 VLAN 的 DHCP 服务器

请求地址

/v1/switch/dhcps/enable/set

请求方法

POST

请求参数示例

```
1 {
2     "vlan_id": 10,
3     "enable": true
4 }
```

请求参数说明

| 参数名称    | 参数类型  | 是否必填 | 描述                              |
|---------|-------|------|---------------------------------|
| vlan_id | int32 | True | VLAN ID                         |
| enable  | bool  | True | DHCP 服务器开关：<br>true-开启，false-关闭 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok"
}
```

4 }

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

12.5.5 获取静态 IP 绑定列表

接口描述

获取静态 IP 绑定列表（手动绑定的客户端）

请求地址

/v1/switch/dhcps/static/get

请求方法

POST

请求参数示例

```
1 {
2     "vlan_id": [10, 20]
3 }
```

请求参数说明

| 参数名称    | 参数类型  | 是否必填  | 描述                 |
|---------|-------|-------|--------------------|
| vlan_id | array | False | VLAN ID 列表，不传则查询所有 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": [
5         {
6             "vlan_id": 1,
7             "ip": "192.168.1.50",
8             "netmask": "255.255.255.0",
9             "type": "Manual",
10            "host_name": "device-001",
11            "mac": "00:11:22:33:44:55",
12            "expired_time": {
13                "days": 0,
```

```

14         "hours": 12,
15         "minutes": 0
16     },
17     "brand": "kiloview",
18     "mode": "DS-2CD2043G2",
19     "sn": "DSGK123456789",
20     "connect_port": 5,
21     "bind_type": 3,
22     "color": "#FF0000"
23 }
24 ]
25 }

```

## 响应参数说明

| 参数名称                      | 参数类型   | 描述                                      |
|---------------------------|--------|-----------------------------------------|
| code                      | int32  | 业务状态码，200 表示成功                          |
| message                   | string | 业务提示信息                                  |
| data                      | array  | 静态 IP 绑定列表                              |
| data.vlan_id              | int32  | 所属 VLAN ID                              |
| data.ip                   | string | 绑定的 IP 地址                               |
| data.netmask              | string | 子网掩码                                    |
| data.type                 | string | 分配类型：Manual（手动绑定）                       |
| data.host_name            | string | 客户端主机名                                  |
| data.mac                  | string | 客户端 MAC 地址                              |
| data.expired_time         | object | 租约剩余时间                                  |
| data.expired_time.days    | int32  | 剩余天数                                    |
| data.expired_time.hours   | int32  | 剩余小时数                                   |
| data.expired_time.minutes | int32  | 剩余分钟数                                   |
| data.brand                | string | 设备品牌                                    |
| data.mode                 | string | 设备型号                                    |
| data.sn                   | string | 设备序列号                                   |
| data.connect_port         | int32  | 连接的交换机端口号                               |
| data.bind_type            | int32  | 绑定类型：0-未绑定，1-IP 绑定，2-MAC 绑定，3-IP+MAC 绑定 |
| data.color                | string | VLAN 颜色                                 |

## 12.5.6 批量设置静态 IP 绑定

### 接口描述

批量设置静态 IP 绑定（IP/MAC 绑定）目前仅支持基于端口绑定

请求地址

/v1/switch/dhcps/static/set

请求方法

POST

请求参数示例

```
1 {
2   "data": [
3     {
4       "vlan_id": 10,
5       "ip": "192.168.10.50",
6       "mac": "00:11:22:33:44:55",
7       "bind_port": 5,
8       "op": 1,
9       "bind_type": 3
10    }
11  ]
12 }
```

请求参数说明

| 参数名称           | 参数类型   | 是否必填  | 描述                                          |
|----------------|--------|-------|---------------------------------------------|
| data           | array  | True  | 静态 IP 绑定列表                                  |
| data.vlan_id   | int32  | True  | VLAN ID                                     |
| data.ip        | string | True  | 要绑定的 IP 地址                                  |
| data.mac       | string | True  | 要绑定的 MAC 地址                                 |
| data.bind_port | int32  | False | 绑 定 的 端 口 号<br>(bind_type 为 1 时使用)          |
| data.op        | int32  | True  | 操作类型：1-绑定，0-解绑                              |
| data.bind_type | int32  | True  | 绑定类型：1:基于端口绑定 2:基于 mac 绑定 3:未绑定 目前仅支持基于端口绑定 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok"
4 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

12.6 电源管理

12.6.1 查询电源信息

接口描述

获取交换机的电源信息等。

请求地址

/v1/switch/power/status/query

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "slot": "slot1",
7       "name": "AVX-KILOLINK-920-A",
8       "status": "Slot Empty",
9       "power": ""
10    },
11    {
12      "slot": "slot2",
13      "name": "AVX-KILOLINK-920-B",
14      "status": "Slot Empty",
15      "power": ""
16    },
17  ]
18 }
```

```
17     {
18         "slot": "PoE",
19         "name": "PoE Total Power(Real-time)",
20         "status": "Power On",
21         "power": "23.77W"
22     }
23 ]
24 }
```

响应参数说明

data 为电源/功耗相关条目数组。

data[] 元素字段说明

| 参数名称    | 参数类型   | 描述                         |
|---------|--------|----------------------------|
| slot    | String | 槽位或逻辑位标识，如 slot1、slot2、PoE |
| name    | String | 模块或统计项名称                   |
| status  | String | 状态描述，如 Slot Empty、Power On |
| power   | String | 功率字符串；无则可为空字符串（如非 PoE 汇总项） |
| 参数名称    | 参数类型   | 描述                         |
| code    | Number | 业务状态码，200 表示成功             |
| message | String | 提示信息                       |
| data    | Array  | 上表为每个 data[] 元素的字段说明       |

12.7 静态路由

12.7.1 添加静态路由

接口描述

添加静态路由

请求地址

/v1/switch/iproutes/add

请求方法

POST

请求参数示例

```
1 {
2   "dest_ip": "192.168.2.0",
3   "dest_mask": 24,
4   "next_hop": "192.168.1.1",
5   "priority": 100
6 }
```

请求参数说明

| 参数名称      | 参数类型   | 是否必填 | 描述                            |
|-----------|--------|------|-------------------------------|
| dest_ip   | string | True | 目标网络地址（如：192.168.2.0）         |
| dest_mask | int32  | True | 目标网络掩码（如：24 表示 255.255.255.0） |
| next_hop  | string | True | 下一跳 IP 地址                     |
| priority  | int32  | True | 路由优先级（数值越小优先级越高）              |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok"
4 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

12.7.2 删除静态路由

接口描述

删除指定静态路由

请求地址

/v1/switch/iproutes/delete

请求方法

POST

请求参数示例

```
1 {
2   "dest_ip": "192.168.2.0",
3   "dest_mask": 24,
4   "next_hop": "192.168.1.1",
5   "priority": 100
6 }
```

请求参数说明

| 参数名称      | 参数类型   | 是否必填 | 描述        |
|-----------|--------|------|-----------|
| dest_ip   | string | True | 目标网络地址    |
| dest_mask | int32  | True | 目标网络掩码    |
| next_hop  | string | True | 下一跳 IP 地址 |
| priority  | int32  | True | 路由优先级     |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok"
4 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

12.7.3 获取全部静态路由列表

接口描述

获取所有静态路由列表

请求地址

/v1/switch/iproutes/get

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "routes": [
5         {
6             "dest_ip": "192.168.2.0",
7             "dest_mask": 24,
8             "next_hop": "192.168.1.1",
9             "priority": 100
10        }
11    ]
12 }
```

响应参数说明

| 参数名称             | 参数类型   | 描述               |
|------------------|--------|------------------|
| code             | int32  | 业务状态码，200 表示成功   |
| message          | string | 业务提示信息           |
| routes           | array  | 静态路由列表           |
| routes.dest_ip   | string | 目标网络地址           |
| routes.dest_mask | int32  | 目标网络掩码位数（如 24）   |
| routes.next_hop  | string | 下一跳 IP 地址        |
| routes.priority  | int32  | 路由优先级（数值越小优先级越高） |

12.8 网络保存

12.8.1 保存网络配置

接口描述

保存网络配置

请求地址

/v1/network/save

请求方法

POST

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述           |
|---------|--------|--------------|
| code    | int    | 状态码，200 表示成功 |
| message | string | 操作结果描述       |
| data    | null   | 无数据返回        |

12.8.2 查询网络配置

接口描述

查询网络配置

请求地址

/v1/network/query

请求方法

GET

请求参数示例

无

请求参数说明

无

响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "status": false
6   }
7 }
```

响应参数说明

| 参数名称        | 参数类型   | 描述           |
|-------------|--------|--------------|
| code        | int    | 状态码，200 表示成功 |
| message     | string | 操作结果描述       |
| data        | object | 系统信息         |
| data.status | bool   | 状态           |

13 告警事件

13.1 告警事件管理

13.1.1 按设备查询告警事件

接口描述

按设备查询告警事件。

请求地址

/v1/alarm/dev/events/get

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型   | 是否必填  | 描述               |
|------|--------|-------|------------------|
| mac  | string | False | MAC 地址（位置：query） |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": [
5         {
6             "id": 1,
7             "status": 1,
8             "alarm_level": 1,
9             "time": "2024-01-01 10:00:00",
10            "message": "设备 CPU 温度过高",
11            "source_port": 0
12        }
13    ]
14 }
```

响应参数说明

| 参数名称             | 参数类型   | 描述                  |
|------------------|--------|---------------------|
| code             | int32  | 业务状态码，200 表示成功      |
| message          | string | 业务提示信息              |
| data             | array  | 设备告警事件列表            |
| data.id          | int32  | 告警事件 ID             |
| data.status      | int32  | 告警状态：1-未读，2-已读      |
| data.alarm_level | int32  | 告警级别：1-错误，2-告警，3-信息 |
| data.time        | string | 首次发生时间              |
| data.message     | string | 告警消息内容              |
| data.source_port | int32  | 来源端口号               |

13.1.2 删除告警事件

接口描述

删除告警事件。

请求地址

/v1/alarm/events/del

请求方法

POST

请求参数示例

```
1 {
2   "ids": [1, 2, 3]
3 }
```

请求参数说明

| 参数名称 | 参数类型  | 是否必填  | 描述                    |
|------|-------|-------|-----------------------|
| ids  | array | False | 要删除的告警事件 ID 列表，支持批量删除 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok"
4 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

13.1.3 分页查询告警事件

接口描述

分页查询告警事件。

请求地址

/v1/alarm/events/get

请求方法

POST

请求参数示例

```
1 {
2   "alarm_level": [1, 2],
```

```
3  "start_time": "2024-01-01 00:00:00",
4  "end_time": "2024-01-31 23:59:59",
5  "status": 1,
6  "page_index": 1,
7  "page_size": 20,
8  "port_id": [1, 2],
9  "keyword": "CPU"
10 }
```

仅分页、不限定其它条件时：

```
1 {
2   "page_index": 1,
3   "page_size": 20
4 }
```

请求参数说明

| 参数名称        | 参数类型         | 是否必填  | 描述                                   |
|-------------|--------------|-------|--------------------------------------|
| alarm_level | array<int32> | False | JSON 键名；告警级别多选。1-错误、2-告警、3-信息        |
| start_time  | string       | False | JSON 键名；开始时间，格式： yyyy-MM-dd HH:mm:ss |
| end_time    | string       | False | JSON 键名；结束时间，格式： yyyy-MM-dd HH:mm:ss |
| status      | int32        | False | JSON 键名；告警状态：1-未读、2-已读               |
| page_index  | int32        | False | JSON 键名；分页页码，建议从 1 开始                |
| page_size   | int32        | False | JSON 键名；每页条数                         |
| port_id     | array<int32> | False | 端口号筛选                                |
| keyword     | string       | False | JSON 键名；模糊搜索告警文案等                    |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "total": 100,
5   "page_index": 1,
6   "page_size": 20,
7   "data": [
8     {
9       "id": 1,
10      "status": 1,
11      "alarm_level": 1,
12      "time": "2024-01-01 10:00:00",
```

```
13         "message": "CPU 使用率超过 80%",
14         "source_port": 1
15     }
16 ]
17 }
```

响应参数说明

| 参数名称             | 参数类型   | 描述                  |
|------------------|--------|---------------------|
| code             | int32  | 业务状态码，200 表示成功      |
| message          | string | 业务提示信息              |
| total            | int32  | 告警事件总数量             |
| page_index       | int32  | 当前分页页码              |
| page_size        | int32  | 每页数据量               |
| data             | array  | 告警事件列表              |
| data.id          | int32  | 告警事件 ID             |
| data.status      | int32  | 告警状态：1-未读，2-已读      |
| data.alarm_level | int32  | 告警级别：1-错误，2-告警，3-信息 |
| data.time        | string | 首次发生时间              |
| data.message     | string | 告警消息内容              |
| data.source_port | int32  | 来源端口号               |

13.1.4 按告警级别筛选事件列表

接口描述

按告警级别筛选事件列表。

请求地址

/v1/alarm/events/levels/get

请求方法

GET

请求参数示例

Query 示例（多级别）：

```
1 /v1/alarm/events/levels/get?alarm_levels=1&alarm_levels=2
```

仅错误级别：

```
1 /v1/alarm/events/levels/get?alarm_levels=1
```

请求参数说明

| 参数名称         | 参数类型         | 是否必填  | 描述                                                  |
|--------------|--------------|-------|-----------------------------------------------------|
| alarm_levels | array<int32> | False | Query 参数名；多选。<br>1-错误、2-告警、3-信息（位置：query，重复键 multi) |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "events_num": 50,
5     "error_total": 10,
6     "warning_total": 30,
7     "info_total": 10,
8     "data": [
9         {
10             "id": 1,
11             "status": 1,
12             "alarm_level": 1,
13             "time": "2024-01-01 10:00:00",
14             "message": "CPU 使用率超过 80%",
15             "source_port": 1
16         }
17     ]
18 }
```

响应参数说明

| 参数名称             | 参数类型   | 描述                  |
|------------------|--------|---------------------|
| code             | int32  | 业务状态码，200 表示成功      |
| message          | string | 业务提示信息              |
| events_num       | int32  | 符合筛选条件的告警事件总数       |
| data             | array  | 告警事件列表              |
| data.id          | int32  | 告警事件 ID             |
| data.status      | int32  | 告警状态：1-未读，2-已读      |
| data.alarm_level | int32  | 告警级别：1-错误，2-告警，3-信息 |
| data.time        | string | 首次发生时间              |
| data.message     | string | 告警消息内容              |
| data.source_port | int32  | 来源端口号               |
| error_total      | int32  | 未读错误级别告警数量          |
| warning_total    | int32  | 未读告警级别告警数量          |
| info_total       | int32  | 未读信息级别告警数量          |

### 13.1.5 批量设置告警事件状态

#### 接口描述

批量设置告警事件状态。

#### 请求地址

/v1/alarm/events/status/set

#### 请求方法

POST

#### 请求参数示例

```
1 {
2     "ids": [1, 2, 3],
3     "status": 2
4 }
```

#### 请求参数说明

| 参数名称   | 参数类型  | 是否必填  | 描述                    |
|--------|-------|-------|-----------------------|
| ids    | array | False | 要处理的告警事件 ID 列表，支持批量操作 |
| status | int32 | False | 目标状态值：1-未读、2-已读       |

#### 响应示例

```
1 {
2     "code": 200,
3     "message": "ok"
4 }
```

#### 响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

### 13.1.6 下载告警导出文件

**接口描述**

下载告警导出的 Excel 表格文件

**请求地址**

/v1/alarm/file/download

**请求方法**

POST

**请求参数示例**

```
1 {
2   "file_path": "alarm_export_xxx.xlsx"
3 }
```

**请求参数说明**

| 参数名称      | 参数类型   | 是否必填 | 描述                                             |
|-----------|--------|------|------------------------------------------------|
| file_path | string | True | 要下载的文件路径（由 AlarmEventsListGet 接口返回的 file_path） |

**响应示例**

直接返回 Excel 文件下载，响应头包含：

- Content-Disposition: attachment; filename=alarm\_export\_xxx.xlsx

**响应参数说明**

| 参数名称 | 参数类型 | 描述         |
|------|------|------------|
| -    | -    | 见上文（文件流响应） |

### 13.1.7 导出告警列表并获取文件路径

**接口描述**

导出告警列表并获取文件路径。

请求地址

/v1/alarm/list/export

请求方法

GET

请求参数示例

```
1 GET /v1/alarm/list/export
2 ?alarm_level=1
3 &alarm_level=2
4 &start_time=2024-01-01 00:00:00
5 &end_time=2024-12-31 23:59:59
6 &status=1
7 &page_index=1
8 &page_size=100
9 &port_id=1
10 &keyword=error
```

请求参数说明

| 参数名称        | 参数类型   | 是否必填  | 描述                          |
|-------------|--------|-------|-----------------------------|
| alarm_level | array  | False | 告警级别筛选，支持多选。1-错误、2-告警、3-信息  |
| start_time  | string | False | 开始时间，格式：yyyy-MM-dd HH:mm:ss |
| end_time    | string | False | 结束时间，格式：yyyy-MM-dd HH:mm:ss |
| status      | int32  | False | 状态筛选：1-未读、2-已读              |
| page_index  | int32  | False | 分页页码                        |
| page_size   | int32  | False | 每页数据量                       |
| port_id     | array  | False | 端口号筛选                       |
| keyword     | string | False | 关键字搜索，模糊匹配告警消息内容            |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
```

```
5     "file_path": "string"
6   }
7 }
```

响应参数说明

| 参数名称           | 参数类型   | 描述                 |
|----------------|--------|--------------------|
| code           | int32  | 业务状态码，200 表示成功     |
| message        | string | 业务提示信息             |
| data           | object | 导出文件信息             |
| data.file_path | string | 导出的告警记录 Excel 文件路径 |

13.2 告警配置

13.2.1 获取告警策略配置

接口描述

获取告警策略配置。

请求地址

/v1/alarm/config/get

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "cpu_enabled": true,
6     "mem_enabled": true,
7     "temp_enabled": true,
8     "alarm_interval": 1,
```

```
9     "disconnect_alarm_enabled": true,
10     "disconnect_duration": 1
11 }
12 }
```

响应参数说明

| 参数名称                          | 参数类型   | 描述                  |
|-------------------------------|--------|---------------------|
| code                          | int32  | 业务状态码，200 表示成功      |
| message                       | string | 业务提示信息              |
| data                          | object | 业务数据主体              |
| data.cpu_enabled              | bool   | CPU 告警开关            |
| data.mem_enabled              | bool   | 内存告警开关              |
| data.temp_enabled             | bool   | 温度告警开关              |
| data.alarm_interval           | int32  | 相同设备的同一类型告警时间间隔（分钟） |
| data.disconnect_alarm_enabled | bool   | 设备意外断连告警开关          |
| data.disconnect_duration      | int32  | 设备意外断连判断时长（秒）       |

13.2.2 设置告警策略配置

接口描述

设置告警策略配置。

请求地址

/v1/alarm/config/set

请求方法

POST

请求参数示例

```
1 {
2     "cpu_enabled": true,
3     "mem_enabled": true,
4     "temp_enabled": true,
5     "alarm_interval": 5,
6     "disconnect_alarm_enabled": true,
7     "disconnect_duration": 30
8 }
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述 |
|------|------|------|----|
|------|------|------|----|

|                          |       |       |                                    |
|--------------------------|-------|-------|------------------------------------|
| cpu_enabled              | bool  | False | CPU 告警开关                           |
| mem_enabled              | bool  | False | 内存告警开关                             |
| temp_enabled             | bool  | False | 温度告警开关                             |
| alarm_interval           | int32 | False | 相同设备的同一类型告警时间间隔（分钟）                |
| disconnect_alarm_enabled | bool  | False | 设备意外断连告警开关                         |
| disconnect_duration      | int32 | False | 设备意外断连判断时长（秒），在设定时间内重新连接上设备则不会触发告警 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok"
4 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

13.3 告警调试

13.3.1 调试：查询全部端口告警（DEBUG）

接口描述

此接口仅用于 DEBUG 调试，返回所有端口的告警事件数据

请求地址

/v1/alarm/ports/events/get

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "1": {
6       "has_alarm": true,
7       "has_error": true,
8       "port_alarm_events": [
9         {
10          "id": 1,
11          "status": 1,
12          "alarm_level": 1,
13          "time": "2024-01-01 10:00:00",
14          "message": "端口 LinkDown",
15          "source_port": 1
16        }
17      ]
18    },
19    "2": {
20      "has_alarm": false,
21      "has_error": false,
22
23      "port_alarm_events": [ ]
24    }
25  }
26 }
27 }
```

响应参数说明

| 参数名称    | 参数类型                       | 描述                              |
|---------|----------------------------|---------------------------------|
| code    | int32                      | 业务状态码，200 表示成功                  |
| message | string                     | 业务提示信息                          |
| data    | map<string, PortAlarmItem> | 端口告警信息，key 为端口号，value 为该端口的告警详情 |

- | data[ ].has\_alarm | bool | 该端口是否有告警事件 |
- | data[ ].has\_error | bool | 该端口是否有错误级别告警 |
- | data[ ].port\_alarm\_events | array | 该端口的告警事件列表 |

### 13.3.2 调试：查询端口告警（DEBUG）

#### 接口描述

此接口仅用于 DEBUG 调试

#### 请求地址

/v1/alarm/debug/port

#### 请求方法

POST

#### 请求参数示例

```
1 {
2   "port_id": 1
3 }
```

#### 请求参数说明

| 参数名称    | 参数类型  | 是否必填  | 描述  |
|---------|-------|-------|-----|
| port_id | int32 | False | 端口号 |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "code": 1,
6     "message": "string",
7     "has_alarm": true,
8     "has_error": true,
9     "alarm_events": [
10      {
11        "id": 1,
12        "status": 1,
13        "alarm_level": 2,
14        "time": "2024-01-01 10:00:00",
15        "message": "端口 LinkDown",
16        "source_port": 1
17      }
18    ],
19     "error_events": [
20      {
21        "id": 2,
22        "status": 1,
23        "alarm_level": 1,
24        "time": "2024-01-01 10:00:00",
25        "message": "端口 Error",
```

```
26         "source_port": 1
27     }
28 ]
29 }
30 }
```

响应参数说明

| 参数名称                     | 参数类型   | 描述                            |
|--------------------------|--------|-------------------------------|
| code                     | int32  | 业务状态码，200 表示成功                |
| message                  | string | 业务提示信息                        |
| has_alarm                | bool   | 该端口是否有告警事件(warning 或 info 级别) |
| has_error                | bool   | 该端口是否有错误级别告警                  |
| alarm_events             | array  | 该端口的告警事件列表(warning 和 info 级别) |
| alarm_events.id          | int32  | 告警事件 ID                       |
| alarm_events.status      | int32  | 告警状态：1-未读，2-已读                |
| alarm_events.alarm_level | int32  | 告警级别：1-错误，2-告警，3-信息           |
| alarm_events.time        | string | 首次发生时间                        |
| alarm_events.message     | string | 告警消息内容                        |
| alarm_events.source_port | int32  | 来源端口号                         |
| error_events             | array  | 该端口的错误级别告警事件列表                |
| error_events.id          | int32  | 告警事件 ID                       |
| error_events.status      | int32  | 告警状态：1-未读，2-已读                |
| error_events.alarm_level | int32  | 告警级别：1-错误，2-告警，3-信息           |
| error_events.time        | string | 首次发生时间                        |
| error_events.message     | string | 告警消息内容                        |
| error_events.source_port | int32  | 来源端口号                         |

14 生产测试

14.1 生产测试与设备信息

14.1.1 恢复出厂设置

接口描述

恢复出厂设置。

请求地址

/v1/test/factory\_reset

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {
5         "code": 1,
6         "message": "string"
7     }
8 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

14.1.2 获取风扇列表

接口描述

获取风扇列表。

请求地址

/v1/test/fan\_list

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     "string"
6   ]
7 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |
| data    | array  | 业务数据主体         |

14.1.3 获取设备资源与版本信息

接口描述

获取设备资源与版本信息。

请求地址

/v1/test/get\_resources

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述 |
|------|------|------|----|
|------|------|------|----|

|     |   |   |       |
|-----|---|---|-------|
| (无) | - | - | 无请求参数 |
|-----|---|---|-------|

## 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "exchange_board": {
6       "cpu_usage": "string",
7       "mem_usage": "string",
8       "temperature": "string",
9       "software_version": "string",
10      "hardware_version": "string"
11    },
12    "ev4": {
13      "cpu_usage": "string",
14      "mem_usage": "string",
15      "temperature": "string",
16      "software_version": "string",
17      "hardware_version": "string"
18    },
19    "a311d2": {
20      "cpu_usage": "string",
21      "mem_usage": "string",
22      "temperature": "string",
23      "software_version": "string",
24      "hardware_version": "string"
25    },
26    "serial_number": "string",
27    "firmware_version": "string",
28    "hardware_version": "string"
29  }
30 }
```

## 响应参数说明

| 参数名称                                 | 参数类型   | 描述             |
|--------------------------------------|--------|----------------|
| code                                 | int32  | 业务状态码，200 表示成功 |
| message                              | string | 业务提示信息         |
| data                                 | object | 业务数据主体         |
| data.exchange_board                  | object | 交换板资源信息        |
| data.exchange_board.cpu_usage        | string | CPU 使用率        |
| data.exchange_board.mem_usage        | string | 内存使用率          |
| data.exchange_board.temperature      | string | 温度             |
| data.exchange_board.software_version | string | 软件版本号          |
| data.exchange_board.hardware_version | string | 硬件版本号          |
| data.ev4                             | object | 4EV 板资源信息      |
| data.ev4.cpu_usage                   | string | CPU 使用率        |
| data.ev4.mem_usage                   | string | 内存使用率          |
| data.ev4.temperature                 | string | 温度             |

|                              |        |              |
|------------------------------|--------|--------------|
| data.ev4.software_version    | string | 软件版本号        |
| data.ev4.hardware_version    | string | 硬件版本号        |
| data.a311d2                  | object | A311D2 板资源信息 |
| data.a311d2.cpu_usage        | string | CPU 使用率      |
| data.a311d2.mem_usage        | string | 内存使用率        |
| data.a311d2.temperature      | string | 温度           |
| data.a311d2.software_version | string | 软件版本号        |
| data.a311d2.hardware_version | string | 硬件版本号        |
| data.serial_number           | string | 设备序列号        |
| data.firmware_version        | string | 固件版本号        |
| data.hardware_version        | string | 硬件版本号        |

### 14.1.4. 查询交换机端口状态

#### 接口描述

查询交换机端口状态。

#### 请求地址

/v1/test/port\_status

#### 请求方法

GET

#### 请求参数示例

```
1 {}
```

#### 请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "port": 1,
7       "speed_duplex": "string",
```

```
8         "link": true,
9         "poe_power": "string",
10        "test_result": "string"
11    }
12 ]
13 }
```

响应参数说明

| 参数名称              | 参数类型   | 描述             |
|-------------------|--------|----------------|
| code              | int32  | 业务状态码，200 表示成功 |
| message           | string | 业务提示信息         |
| data              | array  | 业务数据主体         |
| data.port         | int32  | 返回的端口号         |
| data.speed_duplex | string | 字符串参数          |
| data.link         | bool   | 布尔开关值          |
| data.poe_power    | string | 字符串参数          |
| data.test_result  | string | 返回的返回结果        |

14.1.5 设置序列号

接口描述

设置序列号。

请求地址

/v1/test/set\_serial\_number

请求方法

POST

请求参数示例

```
1 {
2     "serial_number": "SN20240001"
3 }
```

请求参数说明

| 参数名称          | 参数类型   | 是否必填 | 描述    |
|---------------|--------|------|-------|
| serial_number | string | True | 设备序列号 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "code": 1,
6     "message": "string"
7   }
8 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

14.1.6 启动风扇测试

接口描述

启动风扇测试。

请求地址

/v1/test/start\_fan\_test

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "fans": [
6       {
7         "name": "string",
```

```
8         "status": "string",
9         "rpm": 1
10     }
11 ]
12 }
13 }
```

响应参数说明

| 参数名称             | 参数类型   | 描述             |
|------------------|--------|----------------|
| code             | int32  | 业务状态码，200 表示成功 |
| message          | string | 业务提示信息         |
| data             | object | 业务数据主体         |
| data.fans        | array  | 风扇测试结果列表       |
| data.fans.name   | string | 返回的名称          |
| data.fans.status | string | 返回的状态值         |
| data.fans.rpm    | int32  | 数值参数           |

14.1.7 停止风扇测试

接口描述

停止风扇测试。

请求地址

/v1/test/stop\_fan\_test

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {
```

```
5     "code": 1,  
6     "message": "string"  
7 }  
8 }
```

响应参数说明

| 参数名称    | 参数类型   | 描述             |
|---------|--------|----------------|
| code    | int32  | 业务状态码，200 表示成功 |
| message | string | 业务提示信息         |

14.1.8 查询 USB 端口状态

接口描述

查询 USB 端口状态。

请求地址

/v1/test/usb\_port\_status

请求方法

GET

请求参数示例

```
1 {}
```

请求参数说明

| 参数名称 | 参数类型 | 是否必填 | 描述    |
|------|------|------|-------|
| (无)  | -    | -    | 无请求参数 |

响应示例

```
1 {  
2     "code": 200,  
3     "message": "ok",  
4     "data": "string"  
5 }
```

响应参数说明

| 参数名称 | 参数类型  | 描述             |
|------|-------|----------------|
| code | int32 | 业务状态码，200 表示成功 |

|         |        |        |
|---------|--------|--------|
| message | string | 业务提示信息 |
| data    | string | 业务数据主体 |

## 15 AVX 页面接口

### 15.1 获取系统信息

#### 接口描述

获取系统信息

#### 请求地址

/api/systemctrl/report/get\_system\_info

#### 请求方法

GET

#### 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "hostname": "AVX24-4",
6     "version": {
7       "hardwareVersion": "V1.0",
8       "softwareVersion": "V2.0.0",
9       "firmwareVersion": "V1.5.0",
10      "serialnumber": "SN000000000001",
11      "ndiVersion": "NDI 5.5",
12      "product": "AVX24-4",
13      "oem": "Brand"
14    },
15    "mac": "00:11:22:33:44:55",
16    "mode": "L2",
17    "cpu": {
18      "percent": 25
19    },
20    "temperature": 45,
21    "mem": {
22      "percent": 60,
23      "total": 4096,
24      "used": 2458
25    },
26    "disk": {
27      "percent": 30,
28      "total": 10240,
29      "used": 3072
30    }
31  }
```

```
31     }  
32 }
```

响应参数说明

| 参数名称                         | 参数类型   | 描述           |
|------------------------------|--------|--------------|
| code                         | int    | 状态码，200 表示成功 |
| message                      | string | 操作结果描述       |
| data                         | object | 系统信息         |
| data.hostname                | string | 主机名          |
| data.version                 | object | 系统版本         |
| data.version.hardwareVersion | string | 硬件版本         |
| data.version.softwareVersion | string | 软件版本         |
| data.version.firmwareVersion | string | 固件版本         |
| data.version.serialnumber    | string | 序列号          |
| data.version.ndiVersion      | string | NDI 版本       |
| data.version.product         | string | 产品           |
| data.version.oem             | string | OEM          |
| data.mac                     | string | MAC 地址       |
| data.mode                    | string | 模式           |
| data.cpu                     | object | CPU 占用       |
| data.cpu.percent             | int    | CPU 占用率      |
| data.temperature             | int    | 温度           |
| data.mem                     | object | MEM 占用       |
| data.mem.percent             | int    | MEM 占用率      |
| data.mem.total               | int    | MEM 总大小      |
| data.mem.used                | int    | MEM 已用大小     |
| data.disk                    | object | 磁盘占用         |
| data.disk.percent            | int    | 磁盘占用率        |
| data.disk.total              | int    | 磁盘总大小        |
| data.disk.used               | int    | 磁盘已用大小       |

15.2 获取网络接口

接口描述

获取网络接口信息

请求地址

/api/networkmanager/report/get\_network

## 请求方法

GET

## 响应示例

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": [
5     {
6       "name": "eth1.1",
7       "type": "eth",
8       "mac": "00:11:22:33:44:55",
9       "address": "192.168.1.100",
10      "netmask": "255.255.255.0",
11      "gw": "192.168.1.1",
12      "dns": "8.8.8.8",
13      "method": "static",
14      "status": "up",
15      "multicast": "yes",
16      "receive": 102400,
17      "transmit": 51200
18    }
19  ]
20 }
```

## 响应参数说明

| 参数名称             | 参数类型   | 描述                   |
|------------------|--------|----------------------|
| code             | int    | 状态码，200 表示成功         |
| message          | string | 操作结果描述               |
| data             | array  | 网络接口信息               |
| data[].name      | string | 接口名称                 |
| data[].type      | string | 接口类型（固定"eth"）        |
| data[].mac       | string | MAC 地址               |
| data[].address   | string | IP 地址                |
| data[].netmask   | string | 子网掩码                 |
| data[].gw        | string | 网关地址                 |
| data[].dns       | string | DNS 服务器              |
| data[].method    | string | IP 获取方式（DHCP/static） |
| data[].status    | string | 接口状态（up/down）        |
| data[].multicast | string | 是否支持组播               |
| data[].receive   | int    | 接收流量(KB)             |
| data[].transmit  | int    | 发送流量(KB)             |

# 16 AVX 反代设备页面接口

## 16.1 获取子设备 iframe 页面接口

### 接口描述

获取子设备 iframe 页面接口信息

### 请求地址

/v1/local/klnk/forward

### 请求方法

GET

### 请求参数示例

```
1 {
2   "id": "18:C3:F4:21:D7:2E"
3 }
```

### 请求参数说明

| 参数名称 | 参数类型   | 是否必填 | 描述          |
|------|--------|------|-------------|
| id   | string | 是    | 子设备的 MAC 地址 |

### 响应示例

```
1 {
2   "code": 200,
3   "msg": "Success",
4   "requestId": "963f4f61-9079-44dd-82b7-820ce11403a5",
5   "data": [
6     {
7       "serial_number": "201413001D72E",
8       "id": "",
9       "action": "iframeDrawer",
10      "path": "/iframeDrawer/6558d1db07503352fb1211ffee941875/39651/7907/iframeDrawer",
11      "pathParams": {
12        "channel": "",
13        "type": ""
14      },
15      "isWorking": true,
16      "dev_ip": "",
17      "alias": ""
18    }
19  ]
20 }
```

## 响应参数说明

| 参数名称                      | 参数类型   | 描述            |
|---------------------------|--------|---------------|
| code                      | int    | 状态码, 200 表示成功 |
| msg                       | string | 操作结果描述        |
| requestId                 | string | 请求 ID         |
| data                      | array  | 网络接口信息        |
| data[].serial_number      | string | 子设备序列号        |
| data[].id                 | string | 设备 ID         |
| data[].action             | string | 操作的方式         |
| data[].path               | string | 反代路径          |
| data[].pathParams         | object | 反代路径参数        |
| data[].pathParams.channel | string | 通道            |
| data[].pathParams.type    | string | 类型            |
| isWorking                 | bool   | 是否运行正常        |
| dev_ip                    | string | IP 地址         |
| alias                     | string | 别名            |