

MG300V2 网关解码器 API 文档 V1.0.0

文档版本	适配固件
1.0.0	2.20.0132

声明

- 本文档基于 MG300V2 网关 & 解码器固件 2.20.0132 编写，后续固件更新可能新增接口，以最新文档为准。
- 调用 API 时需遵守硬件性能限制，过度调用可能导致设备响应延迟或宕机。
- 未授权的 API 调用（如篡改认证信息、越权访问）可能导致设备异常。

MG300V2系列 网关&解码器 接口

欢迎您使用Device HTTP API!

Device HTTP API是为本型号网络解码和软件开发编程接口。Device HTTP API旨在通过简单的HTTP请求/响应机制，实现对设备的各种功能控制。软件开发者利用Device HTTP API，可以达到远程管理和控制设备的目的。

API的每一个功能，对应有一个HTTP请求路径。在当前版本的API中，这个路径的格式是：[http://\[ip:port/api/xxx\]](http://[ip:port/api/xxx])

本文档将列出每一个API功能对应的文件名(filename)及其功能描述。同时，在每个API文档中，会有一个测试接口，您可以根据请求参数的描述，在测试接口中填写您需要进 行测试的请求参数，执行测试，可以在线验证API功能，十分方便。

API的请求遵循标准的HTTP协议，开发人员可以使用任何符合HTTP标准的工具、软件开发库来完成API调用（例如Web浏览器，JavaScript库，C/C++的HTTP协议库，JAVA，.NET等）。参数的提交根据不同API的接口要求，可能为POST或GET，或者两者均可，具体请参见API说明。API的响应为JSON格式，JSON对象中的每一个参数的意义，请参见每一个API的文档说明。

我们尽可能保持产品API接口的一致，但不可排除因产品特性差异而导致不同产品的API接口会存在部分差异。因此，请开发人员按照官方指导，区别和处理这些细节差异。

目录

- [1. 调用接口说明](#)
 - [1.1 鉴权](#)
 - [1.2 示例](#)
 - [1.3 错误码](#)
- [2. INFO模块](#)
 - [2.1 GET /info/get](#)
- [3. USER模块](#)
 - [3.1 POST /users/login](#)
 - [3.2 GET /users/list](#)
 - [3.3 POST /users/add](#)

- [3.4 POST /users/remove](#)
- [3.5 POST /users/modify](#)
- [3.6 GET /roles/permissions](#)
- [3.7 POST /roles/permupdate](#)
- [3.8 POST /users/logout](#)
- [3.9 POST /users/auth_info_get](#)
- [3.10 POST /users/auth_info_set](#)
- [3.11 GET /users/info](#)
- [3.12 POST /users/session/check](#)
- [3.13 POST /auth/check](#)
- [4. OVERLAY模块](#)
 - [4.1 GET /overlay/list](#)
 - [4.2 POST /overlay/add](#)
 - [4.3 POST /overlay/remove](#)
 - [4.4 POST /overlay/copy](#)
 - [4.5 POST /overlay/rename](#)
 - [4.6 POST /overlay/save](#)
 - [4.7 POST /overlay/cancel](#)
 - [4.8 POST /overlay/select](#)
 - [4.9 POST /overlay/items/set](#)
 - [4.10 GET /overlay/items/get](#)
 - [4.11 POST /overlay/items/add](#)
 - [4.12 POST /overlay/items/remove](#)
 - [4.13 GET /overlay/images/list](#)
 - [4.14 POST /overlay/images/remove](#)
 - [4.15 POST /overlay/images/add](#)
 - [4.16 GET /overlay/editing/get](#)
 - [4.17 POST /overlay/editing/select](#)
 - [4.18 GET /overlay/preview/get](#)
 - [4.19 GET /overlay/datetime/format/list](#)
 - [4.20 GET /overlay/font/list](#)
- [5. SOURCE模块](#)
 - [5.1 GET /source/groups/list](#)
 - [5.2 POST /source/groups/add](#)
 - [5.3 POST /source/groups/streams/add](#)
 - [5.4 POST /source/groups/rename](#)
 - [5.5 POST /source/groups/remove](#)

- [5.6 POST /source/groups/reorder](#)
- [5.7 POST /source/groups/streams/modify](#)
- [5.8 POST /source/groups/streams/remove](#)
- [5.9 POST /source/groups/streams/copy](#)
- [5.10 POST /source/groups/streams/move](#)
- [6. LAYOUT模块](#)
 - [6.1 GET /layout/template/list](#)
 - [6.2 GET /layout/list](#)
 - [6.3 POST /layout/add](#)
 - [6.4 POST /layout/remove](#)
 - [6.5 POST /layout/select](#)
 - [6.6 POST /layout/modify](#)
 - [6.7 POST /layout/save](#)
 - [6.8 POST /layout/resave](#)
 - [6.9 POST /layout/reload](#)
 - [6.10 POST /layout/rename](#)
- [7. OUTPUT模块](#)
 - [7.1 GET /output/interfaces/get](#)
 - [7.2 POST /output/interfaces/set](#)
 - [7.3 GET /output/list](#)
 - [7.4 GET /output/get](#)
 - [7.5 POST /output/source/set](#)
 - [7.6 POST /output/source/remove](#)
 - [7.7 GET /output/resolution/list](#)
 - [7.8 POST /output/resolution/set](#)
 - [7.9 GET /output/audiomix/get](#)
 - [7.10 POST /output/audiomix/set](#)
 - [7.11 POST /output/audiomix/remove](#)
 - [7.12 POST /output/audiomix/add](#)
 - [7.13 POST /ptz/set](#)
 - [7.14 GET /output/audio/sound](#)
 - [7.15 POST /output/border/set](#)
 - [7.16 GET /output/border/get](#)
 - [7.17 POST /output/background/set](#)
 - [7.18 GET /output/background/get](#)
 - [7.19 POST /output/audio/check](#)
 - [7.20 POST /output/ts/program/set](#)

- [7.21 GET /output/ts/program/get](#)
- [7.22 GET /output/encode/get](#)
- [7.23 POST /output/encode/set](#)
- [7.24 POST /output/vumeter/set](#)
- [7.25 POST /output/mute/set](#)
- [8. GATE模块](#)
 - [8.1 GET /gate/stream/list](#)
 - [8.2 POST /gate/stream/add](#)
 - [8.3 POST /gate/stream/delete](#)
 - [8.4 POST /gate/stream/update](#)
 - [8.5 POST /gate/stream/bind_src](#)
 - [8.6 POST /gate/stream/batch_enable](#)
 - [8.7 GET /gate/stream/multi_out_query](#)
 - [8.8 POST /gate/stream/multi_out_enable](#)
- [9. NETWORK模块](#)
 - [9.1 GET /network/get](#)
 - [9.2 POST /network/routing/add](#)
 - [9.3 POST /network/http/set](#)
 - [9.4 GET /network/http/get](#)
 - [9.5 POST /network/ping](#)
 - [9.6 POST /network/routing/set](#)
 - [9.7 POST /network/routing/del](#)
 - [9.8 POST /network/routing/modify](#)
 - [9.9 GET /network/routing/get](#)
 - [9.10 POST /network/set](#)
 - [9.11 POST /network/modify](#)
- [10. SYSTIME模块](#)
 - [10.1 GET /timeregion/settings](#)
 - [10.2 POST /timeregion/time/set](#)
 - [10.3 POST /timeregion/timezone/set](#)
- [11. SYSTEM模块](#)
 - [11.1 GET /sys/reboot](#)
 - [11.2 GET /sys/reset](#)
 - [11.3 GET /sys/restore](#)
 - [11.4 GET /sys/device/get](#)
 - [11.5 POST /sys/device/set](#)
 - [11.6 GET /sys/ip/get](#)

- [11.7 POST /sys/ip/replace](#)
 - [11.8 GET /sys/config/export](#)
 - [11.9 POST /sys/config/import](#)
- [12. MAINTENANCE模块](#)
 - [12.1 GET /maintenance/usage_get](#)
 - [12.2 POST /sys/preset/set](#)
 - [12.3 GET /sys/preset/get](#)
- [13. FIRMWARE模块](#)
 - [13.1 POST /firmware/upgrade](#)
- [14. PREVIEW模块](#)
 - [14.1 GET /preview/get](#)
 - [14.2 POST /preview/source/modify](#)
 - [14.3 POST /preview/source/remove](#)
- [15. NDI-Discovery模块](#)
 - [15.1 GET /ndi/discovery/all](#)
 - [15.2 POST /ndi/discovery/auto/del](#)
 - [15.3 GET /ndi/discovery/sources](#)
 - [15.4 POST /ndi/discovery/manual/add](#)
 - [15.5 POST /ndi/discovery/auto/add](#)
 - [15.6 POST /ndi/discovery/server/del](#)
 - [15.7 POST /ndi/discovery/server/add](#)

1. 调用接口说明

统一接口前缀：

Base URLs: <http://ip:port/api/xxx>

所有接口的返回形式都统一为：

- 正常返回 `{"result": "ok", "data": 某种类型的数据, 比如字符串、数值、数组、对象等等, 可能没有这个字段}`
无data字段返回示例: `{"result": "ok"}`
- 错误返回 `{"result": "error", "msg": "errCode"}`

关于错误返回（重要）：

- 所有接口在失败时都返回统一结构 `{"result": "error", "msg": "<错误码>"}`，其中 `msg` 是「错误码」表中数字错误码的**字符串**形式（例如 `"result": "error", "msg": "100"`）。本文档正文各接口的“返回示例”通常只给出成功（`result: "ok"`）的结构，未逐一列出错误分支，集成时请统一按上述错误结构处理，并参照下方「错误码」表判断具体原因。
- 少数内部/klnk 接口的错误结构可能附带额外的 `code` 字段
(`{"result": "error", "msg": "...", "code": ...}`)，非本文档面向的标准接口，可忽略。

关于 HTTP 方法： 本文档标注的 GET/POST 表示接口的**语义约定**，服务端当前未对请求方法做强制校验（GET/POST 均可达）。请仍按标注方法调用；尤其 `/sys/reboot`、`/sys/reset`、`/sys/restore` 等会重启/重置设备，请谨慎触发。

1.1 鉴权

设备 API 采用基于令牌（Token）的认证机制。除登录接口外，所有 API 请求均需在 HTTP 请求头中携带认证信息。

认证流程：

- 获取令牌：** 客户端调用 `/users/login` 接口，使用用户名和密码进行认证，成功后服务端返回一个会话令牌（token）。
- 携带令牌：** 在后续所有 API 请求中，将令牌以 JSON 字符串格式放入名为 `app` 的 HTTP 请求头（header）中。
- 令牌失效：** 调用 `/users/logout` 注销登录后令牌立即失效。设备重启或长时间不活动也可能导致令牌过期，此时需重新登录获取新令牌。

请求头格式：

参数名	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	JSON 字符串，格式为 <code>{"token":"xxx"}</code> 。登录接口无需携带此头部

app 头 JSON 格式详解：

```
{
  "token": "3515d2e8d3615fc68c84f00b361c5d8c"
}
```

- `token`：登录成功后服务端返回的会话令牌（32 位十六进制字符串）。服务端根据该字段判断会话是否有效。
- 整个 JSON 对象需作为字符串值放入 `app` 请求头中。注意 JSON 字符串中的双引号需要转义或直接作为 HTTP 头值传递（大多数 HTTP 客户端会自动处理）。
- `token` 缺失或无效时返回鉴权错误（错误码 2）。

不需要鉴权的接口：

以下接口无需在请求头中携带 `app`：

- `POST /users/login`（登录接口本身）
- `GET /info/get`（获取设备基本信息，通常用于初始化阶段获取设备身份）

1.2 示例

- (1) API 认证原理
- (2) 登录获取令牌

当调用登录 API（通过 GET/POST 方式）时，响应会包含一个 token。该令牌将作为后续 API 请求的认证凭证。

(3) 在请求中使用令牌

对于其他任何 API 端点，必须在 HTTP 请求头中添加名为 `app` 的鉴权字段，值为 JSON 字符串，格式如下：

```
{
  "token": "3515d2e8d3615fc68c84f00b361c5d8c"
}
```

curl 示例：

```
curl -X GET "http://ip:port/api/info/get" \
-H 'app: {"token":"3515d2e8d3615fc68c84f00b361c5d8c"}'
```

令牌的作用是向服务器证明已通过认证，原理类似 Web 图形界面通过 Cookie 维持会话状态。

(4) 本地演示步骤（含 Postman 操作）

步骤 1：登录获取令牌

- 打开 Postman，新建 POST 请求。
- 请求地址：<https://192.168.28.30/api/users/login>
- 请求体（JSON 格式）：`{ "username": "admin", "password": "your-device-password" }`
(注：若设备未修改密码，默认密码为 admin。)
- 发送请求：点击 Send，从响应中提取 token 字段（如 3515d2e8d3615fc68c84f00b361c5d8c）。

步骤 2：在请求中添加认证头部

- 新建目标 API 请求（如获取信息的 `info/get.json`）。
- 设置请求头：
 - 切换至 Headers 标签页，添加以下键值对：
 - Key: `app`
 - Value: `{"token":"your-obtained-token"}`
(注：将 `your-obtained-token` 替换为步骤 1 中实际获取的令牌。)
- 发送请求：点击 Send，服务器通过令牌验证身份后返回数据。

(5) 浏览器开发者工具获取令牌（备选方法）

- 浏览器按 F12 打开开发者工具，切换至 Network 标签页。

• 执行登录操作，找到登录请求（如 `/users/login`），在响应体的 `data.token` 字段中复制令牌。登录成功时响应形如：

```
{
  "data": {
    "username": "admin",
    "token": "3515d2e8d3615fc68c84f00b361c5d8c",
    "role": "admin",
    "alias": "超级管理员"
  },
  "result": "ok"
}
```

- 将 token 填入后续请求的 app 头即可。

(6) 故障排除提示

- 密码验证：确保设备密码正确（未修改时默认密码为 admin）。
- 令牌有效性：检查令牌是否复制正确，且未过期（过期需重新登录获取）。
- 请求头格式：确认 app 头部的 JSON 格式正确，令牌字段无拼写错误（如 token 需与服务器要求一致）。

1.3 错误码

错误码	错误描述	枚举名（代码）
0	请求成功	ERR_OK
1	参数错误	ERR_PARAMS
2	鉴权错误（token 无效）	ERR_USER_TOKEN_INVALID
100	输出未找到	ERR_OUTPUT_NOT_FOUND
101	视频源未找到	ERR_VIDEO_NOT_FOUND
102	音频源未找到	ERR_AUDIO_NOT_FOUND
103	输出接口未找到	ERR_INTERFACE_NOT_FOUND
104	布局模板未找到	ERR_TEMPLATE_NOT_FOUND
105	布局未找到	ERR_LAYOUT_NOT_FOUND
106	预览位置错误	ERR_PREVIEW_POSITION
107	布局超出范围	ERR_LAYOUT_EXCEEDS_LIMIT
200	组未找到	ERR_GROUP_NOT_FOUND
201	流未找到	ERR_STREAM_NOT_FOUND
202	流不存在	ERR_STREAM_NOT_EXIST
203	NDI服务器重复	ERR_REPEAT_NDI_SERVER
204	流重复	ERR_REPEAT_STREAM

错误码	错误描述	枚举名（代码）
205	不支持的协议	ERR_NOT_SUPPORT_PROTOCOL
206	分组已存在	ERR_GROUP_NAME_DUPLICATION
207	NDI缓存超出最大限制	ERR_NDI_BUFF_NOT_SUPPORT_LIMIT
300	用户未找到	ERR_USER_NOT_FOUND
301	密码不匹配	ERR_PASSWORD_MISMATCH
302	用户名或密码错误	ERR_PWD_FAIL
400	未ping通	ERR_IP_UNREACHABLE
401	网络配置修改失败	ERR_MODIFY_NET
402	路由重复	ERR_REPEAT_ROUTE
403	路由未找到	ERR_ROUTE_NOT_FOUND
404	路由不存在	ERR_ROUTE_NOT_EXIST
405	端口已被占用	ERR_PORT_ALREADY_OCCUPIED
500	名称重复	ERR_REPEAT_NAME
501	正在使用	ERR_IN_USE
502	保存配置失败	ERR_SAVE_FILE
503	时间/时区设置失败	ERR_TIME_ZONE
504	无效的VO输出	ERR_INVALID_VO
505	混音器只支持四路音频	ERR_INVALID_AUDIO_MIXER
506	目标已存在	ERR_TARGET_EXIST
507	文件不存在	ERR_FILE_NOT_EXIST
508	PTZ设置失败	ERR_PTZ_SET_FAILED
509	设备性能不足	ERR_PERFORMANCE_LIMIT_DEVICE
510	设备性能恢复正常	ERR_PERFORMANCE_NORMAL
511	预览数量超出限制	ERR_PREVIEW_OVER_NUMBER
608	网关端口被占用	ERR_GATEWAY_PORT_ALREADY_OCCUPIED
609	网关流信息未找到	ERR_GATEWAY_STREAM_NOT_FOUND
610	网关推送未绑定源	ERR_GATEWAY_STREAM_NOT_BIND_SRC
611	1个源最多允许推送4路输出	ERR_GATEWAY_STREAM_MAX_4_OUTS_1_SRC

错误码	错误描述	枚举名（代码）
612	最多允许推送36路输出	ERR_GATEWAY_STREAM_MAX_36_OUTS_9_SRCS
613	最多允许推送9个源	ERR_GATEWAY_STREAM_MAX_9_SRCS
614	NDI_HX通道名重复	ERR_GATEWAY_NDI_CHANNEL_EXISTS
615	HLS会话ID重复	ERR_GATEWAY_HLS_SESSION_EXISTS
616	多画面预览绑定失败	ERR_GATEWAY_BIND_MULTIVIEW_FAILED
617	多画面编码参数无效	ERR_MULTIVIEW_ENCODE_ARGS_INVALID
800	拉流连接失败	ERR_PULL_STREAM_CONNECT_FAILED
801	拉流认证失败	ERR_PULL_STREAM_AUTH_FAILED
802	获取流信息失败	ERR_PULL_STREAM_FIND_INFO_FAILED
803	流信息中未找到音频或视频轨道	ERR_PULL_STREAM_NOT_FOUND_TRACK
804	分辨率不支持	ERR_PULL_STREAM_RESOLUTION_NOT_SUPPORT
805	连接尝试超过最大次数	ERR_PULL_STREAM_TRY_REACH_MAX_TIMES
1000	视频解码失败	ERR_VIDEO_DECODE_ERROR
-1000	不支持10bit的H264/H265/HX解码	ERR_PULL_STREAM_NOT_SUPPORT_10BIT_VIDEO

2. INFO模块

INFO（设备信息）模块用于获取设备的基本信息，包括设备名称、固件版本、硬件版本、序列号和软件版本等。通常在客户端初始化时调用，用于确认设备身份和固件版本兼容性。

2.1 GET /info/get

功能说明

获取设备基本信息，包括设备名称、固件版本、硬件版本、序列号和软件版本。用于查询设备身份和版本信息，通常在客户端初始化时调用。此接口无需携带鉴权令牌即可访问。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	当前用户的认证令牌，格式为JSON字符串 <code>{"token":"xxx"}</code> 。此接口可不携带

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none	返回数据	设备基本信息对象
»» device_name	string	true	none	设备名称	设备的自定义名称
»» firmware_version	string	true	none	固件版本	当前运行的固件版本号
»» hardware_version	string	true	none	硬件版本	设备硬件版本标识
»» serial_number	string	true	none	设备序列号	设备出厂唯一序列号
»» software_version	string	true	none	软件版本	应用软件版本号
» result	string	true	none	返回结果	固定返回 "ok" 表示请求成功

3. USER模块

权限分级模块，管理用户账户与角色权限。

预设角色定义

系统内置 5 种预设角色，role_id 固定不可变：

role_id	角色标识	角色名	说明
1	admin	超级管理员	系统超级管理员，拥有全部权限，管理所有角色与用户
2	admin_regular	普通管理员	由超级管理员创建，可管理普通用户，不可管理其他管理员
3	General Role 1	通用角色1	固定角色，权限由管理员逐项配置
4	General Role 2	通用角色2	固定角色，权限由管理员逐项配置
5	General Role 3	通用角色3	固定角色，权限由管理员逐项配置

预设权限定义

权限节点采用树形层级结构，id 固定不可变。is_group 为 true 的节点为分组容器，其下包含 children 子节点。

id	权限标识	显示名	父分组	is_group
1	ID_PERM_HOME	主页	—	true
2	ID_PERM_SOURCE	源列表	主页	false
3	ID_PERM_PREVIEW	预览列表	主页	true
4	ID_PERM_ENCODE	编码设置	预览列表	false
5	ID_PERM_PREVIEW_MANAGE	预览管理	预览列表	false
6	ID_PERM_OUTPUT_IF	物理接口输出配置	主页	true

id	权限标识	显示名	父分组	is_group
7	ID_PERM_RESOLUTION	分辨率帧率	物理接口输出配置	false
8	ID_PERM_AUDIOMIX	调音台	物理接口输出配置	false
9	ID_PERM_AUDIO_OUT_EDIT	音频输出编辑	物理接口输出配置	false
10	ID_PERM_VIDEO_OUT_EDIT	视频输出编辑	物理接口输出配置	false
11	ID_PERM_LAYOUT	多画面显示配置	主页	true
12	ID_PERM_LAYOUT_SELECT	布局选择	多画面显示配置	false
13	ID_PERM_LAYOUT_EDITOR	布局编辑器	多画面显示配置	false
14	ID_PERM_LAYOUT_TOOLS	其它工具	多画面显示配置	false
15	ID_PERM_LAYOUT_OVERLAY	叠层编辑器	多画面显示配置	false
16	ID_PERM_GATE_SERVICE	流服务	—	true
17	ID_PERM_GATE_STREAM	流服务查看权限	流服务	false
18	ID_PERM_GATE_BATCH	流服务启停	流服务查看权限	false
19	ID_PERM_GATE_BIND_SRC	流服务推流源配置	流服务查看权限	false
20	ID_PERM_SETTINGS	设置	—	true
21	ID_PERM_SYS_DEVICE	通用设置	设置	false
22	ID_PERM_NETWORK	网络设置	设置	false
23	ID_PERM_USERS	用户列表	设置	false
24	ID_PERM_FIRMWARE	固件升级	设置	false
25	ID_PERM_MAINTENANCE	高级设置	设置	false
26	ID_PERM_SYS_PRESET	预设配置	设置	false
27	ID_PERM_SYS_RESET_GROUP	系统重置	设置	true
28	ID_PERM_SYS_RESET	恢复出厂	系统重置	false
29	ID_PERM_SYS_REBOOT	设备重启	系统重置	false
30	ID_PERM_LOG	系统日志	设置	false
31	ID_PERM_ROLES	角色权限管理	设置	false
32	ID_PERM_USERS_EDIT	用户编辑	设置	false

权限错误码定义

权限校验失败时返回的错误码，枚举值 `PermErrorCode`，从 2000 自增，对应 `msg` 字段。

值	枚举名	中文	英文	韩文
2000	PERM_ERR_USERNAME_INVALID	用户名格式 不合法	Invalid username format	유효하지 않은 사용자 이름 형 식
2001	PERM_ERR_ROLE_NOT_FOUND	角色不存在	Role not found	역할을 찾을 수 없음
2002	PERM_ERR_DENIED	权限校验失 败	Permission denied	권한 거부됨
2003	PERM_ERR_CANNOT_DELETE_SELF	不能删除自 己	Cannot delete self	자신을 삭제할 수 없음
2004	PERM_ERR_CANNOT_MODIFY_PRESET	不能修改预 置角色/权限	Cannot modify preset role/permission	사전 설정된 역 할/권한을 수정 할 수 없음
2005	PERM_ERR_USER_LIMIT_REACHED	用户数达上 限 (32)	User limit reached (32)	사용자 한도 도 달 (32)
2006	PERM_ERR_HOME	主页	Home	홈
2007	PERM_ERR_SOURCE	无源列表管 理权限	No source list permission	소스 목록 관리 권한 없음
2008	PERM_ERR_PREVIEW	无预览列表 管理权限	No preview list permission	미리보기 목록 관리 권한 없음
2009	PERM_ERR_ENCODE	无编码设置 权限	No encode settings permission	인코딩 설정 권 한 없음
2010	PERM_ERR_OUTPUT_IF	无物理接口 输出配置权 限	No output interface permission	물리적 인터페 이스 출력 구성 권한 없음
2011	PERM_ERR_RESOLUTION	无分辨率、 帧率设置权 限	No resolution/framerate permission	해상도/프레임 속도 설정 권한 없음
2012	PERM_ERR_AUDIOMIX	无调音台权 限	No audio mixer permission	오디오 믹서 권 한 없음
2013	PERM_ERR_AUDIO_OUT_EDIT	无音频输出 设置权限	No audio output settings permission	오디오 출력 설 정 권한 없음
2017	PERM_ERR_VIDEO_OUT_EDIT	无视频输出 设置权限	No video output settings permission	비디오 출력 설 정 권한 없음
2019	PERM_ERR_LAYOUT	无多画面显 示配置权限	No layout configuration permission	멀티뷰 레이아 웃 구성 권한 없음
2020	PERM_ERR_LAYOUT_SELECT	无布局选择 权限	No layout selection permission	레이아웃 선택 권한 없음

值	枚举名	中文	英文	韩文
2021	PERM_ERR_LAYOUT_EDITOR	无布局管理权限	No layout editor permission	레이아웃 편집기 권한 없음
2027	PERM_ERR_LAYOUT_TOOLS	无其它工具使用权限	No tools permission	기타 도구 사용 권한 없음
2034	PERM_ERR_LAYOUT_OVERLAY	无叠层编辑器权限	No overlay editor permission	오버레이 편집기 권한 없음
2035	PERM_ERR_GATE_SERVICE	无流服务管理权限	No streaming service permission	스트리밍 서비스 관리 권한 없음
2036	PERM_ERR_GATE_STREAM	无流服务查看权限	No stream view permission	스트림 보기 권한 없음
2037	PERM_ERR_GATE_BATCH	无流服务启停权限	No stream start/stop permission	스트림 시작/중지 권한 없음
2038	PERM_ERR_GATE_BIND_SRC	无推流源配置权限	No stream source binding permission	스트림 소스 바인딩 권한 없음
2039	PERM_ERR_SETTINGS	无系统设置权限	No system settings permission	시스템 설정 권한 없음
2040	PERM_ERR_SYS_DEVICE	无通用设置权限	No device settings permission	장치 설정 권한 없음
2041	PERM_ERR_NETWORK	无网络设置权限	No network settings permission	네트워크 설정 권한 없음
2042	PERM_ERR_USERS	无用户列表管理权限	No user management permission	사용자 관리 권한 없음
2043	PERM_ERR_FIRMWARE	无固件升级权限	No firmware upgrade permission	펌웨어 업그레이드 권한 없음
2044	PERM_ERR_MAINTENANCE	无高级设置权限	No maintenance settings permission	고급 설정 권한 없음
2045	PERM_ERR_SYS_PRESET	无预设配置权限	No preset configuration permission	사전 설정 구성 권한 없음
2046	PERM_ERR_SYS_RESET_GROUP	无系统重置权限	No system reset permission	시스템 재설정 권한 없음
2047	PERM_ERR_SYS_RESET	无恢复出厂权限	No factory reset permission	공장 초기화 권한 없음
2048	PERM_ERR_SYS_REBOOT	无设备重启权限	No device reboot permission	장치 재부팅 권한 없음
2049	PERM_ERR_LOG	无系统日志管理权限	No system log permission	시스템 로그 관리 권한 없음

值	枚举名	中文	英文	韩文
2050	PERM_ERR_ROLES	无角色管理权限	No role management permission	역할 관리 권한 없음
2051	PERM_ERR_USERS_EDIT	无用户编辑权限	No user edit permission	사용자 편집 권한 없음

返回示例

```
{
  "data": {
    "device_name": "xxx",
    "firmware_version": "xxx",
    "hardware_version": "xxx",
    "serial_number": "xxx",
    "software_version": "2.0.0"
  },
  "result": "ok"
}
```

3.1 POST /users/login

功能说明

用户登录接口。使用用户名和密码进行认证，成功后返回会话令牌（token）及用户完整信息。该 token 须在后续所有 API 请求的 `app` 头中携带，格式为 `{"token":"xxx"}`。token 具有会话生命周期，设备重启或长时间不活动将失效，届时需重新登录获取新令牌。系统支持五种预设角色：超级管理员（admin）、普通管理员（admin_regular）以及三个通用角色（gen_role_1/2/3）。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	登录接口无需携带 token，此头部可省略
body	body	object	否	请求体	包含登录凭证的 JSON 对象
» username	body	string	是	用户名	登录用户名
» password	body	string	是	密码	登录密码

Body 请求参数

超级管理员

```
{
  "username": "admin",
  "password": "admin"
}
```

普通管理员

```
{
  "username": "admin1",
  "password": "admin1"
}
```

普通用户

```
{
  "username": "gen1",
  "password": "gen1"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	none	返回数据	登录成功返回的用户完整信息
»» alias	string	true	none	别名	用户别名
»» changed	boolean	true	none	密码状态	是否修改过密码，true 表示已修改过默认密码
»» create_time	string	true	none	创建时间	用户创建时间戳（Unix 时间戳字符串）
»» result	string	true	none	返回结果	固定返回 "ok" 表示登录成功
»» role	string	true	none	角色标识	用户所属角色标识名，如 admin / admin_regular / gen_role_1 等
»» role_id	integer	true	none	角色ID	关联角色权限表的唯一标识
»» token	string	true	none	验证 token	会话令牌，后续请求需在 app 头中携带，格式为 {"token": "xxx"}
»» user_id	integer	true	none	用户ID	用户唯一标识
»» username	string	true	none	账号名称	登录用户名
» result	string	true	none	返回结果	固定返回 "ok" 表示请求成功

用户角色对应关系见 [预设角色定义](#)。

返回示例

超级管理员

```
{
  "data": {
    "alias": "超级管理员",
    "changed": true,
    "create_time": "169665313",
    "result": "ok",
    "role": "admin",
    "role_id": 1,
    "token": "901fcec0640576e2002e52e32eb8022",
    "user_id": 1,
    "username": "admin"
  },
  "result": "ok"
}
```

普通管理员

```
{
  "data": {
    "alias": "test1",
    "changed": true,
    "create_time": "169733754",
    "result": "ok",
    "role": "admin_regular",
    "role_id": 2,
    "token": "9b8b92fb007a86dee14f5d89bf0c4d2a",
    "user_id": 2,
    "username": "admin1"
  },
  "result": "ok"
}
```

普通用户

```
{
  "data": {
    "alias": "test1",
    "changed": true,
    "create_time": "84917946",
    "result": "ok",
    "role": "gen_role_1",
    "role_id": 3,
    "token": "973429fbd838ccabef4bc76631eab332",
    "user_id": 3,
    "username": "gen1"
  },
  "result": "ok"
}
```

3.2 GET /users/list

功能说明

获取所有用户列表及可用角色信息。返回系统中全部用户的详细信息（用户名、别名、角色、创建时间、电话等），同时返回系统中所有可用角色的列表。支持通过 `search` 参数按用户名、别名或角色名进行过滤查询，便于在用户较多时快速定位。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 <code>{"token": "xxx"}</code>
search	body/query	string	否	搜索关键词	可选，按用户名、别名或角色名过滤结果

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
result	string	true	none	返回结果	固定返回 <code>"ok"</code> 表示请求成功
» data	object	true	none	返回数据	包含用户列表和角色列表的数据对象
»» count	integer	true	none	用户总数	当前系统中的用户总数量
»» list	[object]	true	none	用户列表	用户详细信息列表
»»» alias	string	false	none	别名	用户别名
»»» changed	boolean	false	none	密码状态	是否修改过密码，true 表示已修改过默认密码
»»» create_time	string	false	none	创建时间	用户创建时间戳（Unix 时间戳字符串）
»»» created_by	integer	false	none	创建者 ID	创建该用户的 user_id
»»» descript	string	false	none	描述	用户详细描述
»»» id	integer	false	none	用户 ID	用户唯一标识

名称	类型	必选	约束	中文名	说明
»»» is_self	boolean	false	none	是否自己	是否为当前登录用户本人
»»» is_system	boolean	false	none	系统角色	是否为系统管理员角色（admin 或 admin_regular）
»»» phone	string	false	none	电话	用户联系电话号码
»»» role_id	integer	false	none	角色ID	关联角色权限表的唯一标识
»»» role_name	string	false	none	角色名	用户所属角色标识名称
»»» status	integer	false	none	状态	用户状态，1 表示启用
»»» token	string	false	none	令牌	用户当前的会话令牌值
»»» updated_at	string	false	none	更新时间	用户信息的最近更新时间
»»» username	string	false	none	用户名	用户登录用户名
»» roles	[object]	true	none	角色列表	系统中所有可用角色的信息
»»» created_by	integer	false	none	创建者ID	创建该角色的 user_id
»»» description	string	false	none	描述	角色描述信息
»»» id	integer	false	none	角色ID	角色唯一标识
»»» is_system	boolean	false	none	系统角色	是否为系统固定角色
»»» label	string	false	none	角色标签	角色显示标签/名称
»»» name	string	false	none	角色名	角色标识名称
»»» status	integer	false	none	状态	角色状态，1 表示启用

返回示例

```
{
  "data": {
    "count": 3,
    "list": [
      {
        "alias": "test1",
        "changed": true,
```

```
"create_time": "169733754",
"created_by": 1,
"descript": "",
"id": 2,
"is_self": false,
"is_system": true,
"phone": "",
"role_id": 2,
"role_name": "admin_regular",
"status": 1,
"token": "9b8b92fb007a86dee14f5d89bf0c4d2a",
"updated_at": "1970-01-03T07:08:53",
"username": "admin1"
},
{
  "alias": "test1",
  "changed": true,
  "create_time": "169788417",
  "created_by": 2,
  "descript": "",
  "id": 3,
  "is_self": false,
  "is_system": false,
  "phone": "",
  "role_id": 3,
  "role_name": "gen_role_1",
  "status": 1,
  "token": "e5d25b942c969c1697f57c8777c440bf",
  "updated_at": "1970-01-03T07:09:48",
  "username": "gen1"
},
{
  "alias": "test2",
  "changed": true,
  "create_time": "169793362",
  "created_by": 2,
  "descript": "",
  "id": 4,
  "is_self": false,
  "is_system": false,
  "phone": "",
  "role_id": 5,
  "role_name": "gen_role_3",
  "status": 1,
  "token": "",
  "updated_at": "1970-01-03T07:09:53",
  "username": "gen2"
}
],
"roles": [
  {
    "created_by": 2,
    "description": "固定角色，权限由管理员逐项配置",
```

```
    "id": 3,
    "is_system": false,
    "label": "gen_role_1",
    "name": "gen_role_1",
    "status": 1
  },
  {
    "created_by": 2,
    "description": "固定角色，权限由管理员逐项配置",
    "id": 4,
    "is_system": false,
    "label": "通用角色2",
    "name": "gen_role_2",
    "status": 1
  },
  {
    "created_by": 2,
    "description": "固定角色，权限由管理员逐项配置",
    "id": 5,
    "is_system": false,
    "label": "通用角色3",
    "name": "gen_role_3",
    "status": 1
  }
]
},
"result": "ok"
}
```

3.3 POST /users/add

功能说明

添加新用户。通过 `role_id` 指定用户所属角色（参见预设角色定义表），需要提供用户名、密码和别名。可选提供联系电话和描述信息。仅管理员及以上角色可执行此操作，且用户总数受系统上限（32 个）限制。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 <code>{"token": "xxx"}</code>
body	body	object	否	请求体	包含新用户信息的 JSON 对象
» username	body	string	是	用户名	新用户的登录用户名
» password	body	string	是	密码	新用户的登录密码

名称	位置	类型	必选	中文名	说明
» alias	body	string	是	别名	新用户的显示别名
» role_id	body	integer	是	角色ID	对应角色权限表中的 role_id (1-5)
» phone	body	string	否	联系电话	新用户的联系电话
» descript	body	string	否	描述	新用户的描述信息

Body 请求参数

普通管理员

```
{
  "username": "admin1",
  "password": "admin1",
  "alias": "test1",
  "role_id": 2,
  "phone": "",
  "descript": ""
}
```

普通用户1

```
{
  "username": "gen1",
  "password": "gen1",
  "alias": "test1",
  "role_id": 3,
  "phone": "",
  "descript": ""
}
```

普通用户2

```
{
  "username": "gen2",
  "password": "gen2",
  "alias": "test2",
  "role_id": 5,
  "phone": "",
  "descript": ""
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none	返回结果	固定返回 <code>"ok"</code> 表示用户添加成功

返回示例

```
{
  "result": "ok"
}
```

3.4 POST /users/remove

功能说明

删除指定用户。该操作不可撤销，被删除用户的会话令牌将立即失效。admin 用户（超级管理员）不可被删除。仅管理员及以上角色可执行此操作，且不能删除自身。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 <code>{"token": "xxx"}</code>
body	body	object	否	请求体	包含要删除用户 ID 的 JSON 对象
» id	body	integer	是	用户ID	要删除的用户的唯一标识

Body 请求参数

```
{
  "id": 3
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none	返回结果	固定返回 <code>"ok"</code> 表示用户删除成功

返回示例

```
{
  "result": "ok"
}
```

3.5 POST /users/modify

功能说明

修改用户信息。服务端按 `id` 定位目标用户，仅更新请求中传入的字段。除 `id` 外所有字段均为可选，传入哪个字段就更新哪个字段，未传入的字段保持不变。可修改用户名、密码、别名、角色、电话和描述。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 <code>{"token": "xxx"}</code>
body	body	object	否	请求体	包含要修改字段的 JSON 对象
» id	body	integer	是	用户ID	要修改的用户的唯一标识
» username	body	string	否	用户名	新的登录用户名，不传则不修改
» password	body	string	否	密码	新的登录密码，不传则不修改
» alias	body	string	否	别名	新的用户别名，不传则不修改
» role_id	body	integer	否	角色ID	新的角色ID，关联角色权限表，不传则不修改
» phone	body	string	否	电话	新的联系电话，不传则不修改
» descript	body	string	否	描述	新的用户描述，不传则不修改

Body 请求参数

```
{
  "id": 3,
  "username": "gen1",
  "password": "gen1",
  "alias": "gen1",
  "role_id": 3,
  "phone": "124666",
  "descript": "12312321323"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none	返回结果	固定返回 "ok" 表示用户信息修改成功

返回示例

```
{
  "result": "ok"
}
```

3.6 GET /roles/permissions

功能说明

获取所有角色及其权限列表。返回每个角色的详细信息，包含层级化的权限树结构。权限树中 is_group 为 true 的节点为分组容器节点，其 children 数组包含下级权限节点；is_group 为 false 的节点为叶子权限节点。checked 字段标记该权限项当前是否已启用。此接口用于角色权限管理页面的数据查询。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 {"token": "xxx"}

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none	返回结果	固定返回 "ok" 表示请求成功
» data	[object]	true	none	角色列表	所有角色及其权限的数组
»» created_by	integer	false	none	创建者 ID	创建该角色的 user_id
»» description	string	false	none	描述	角色描述信息
»» id	integer	true	none	角色ID	角色唯一标识
»» is_self	boolean	true	none	自身角色	是否为当前登录用户所属的角色

名称	类型	必选	约束	中文名	说明
»» is_system	boolean	true	none	系统角色	是否为系统固定角色（不可删除）
»» label	string	true	none	显示名	角色显示名称
»» name	string	true	none	角色名	角色标识名称
»» permissions	[object]	true	none	权限树	层级化权限节点列表
»»» children	[object]	false	none	子节点	下级权限节点数组
»»»» checked	boolean	false	none	已勾选	该权限项是否已启用
»»»» id	integer	false	none	权限ID	权限项唯一标识
»»»» is_group	boolean	false	none	分组节点	true 表示分组容器，false 表示叶子权限节点
»»»» label	string	false	none	权限名	权限项显示名称
»»»» multiple_selected	boolean	false	none	部分选中	分组节点下存在部分选中的子节点时为true
»»»» translate_name	string	false	none	权限标识	权限英文标识名
»»» checked	boolean	false	none	已勾选	顶层分组节点无此字段，子节点标记是否启用
»»» id	integer	true	none	权限ID	权限项唯一标识
»»» is_group	boolean	true	none	分组节点	true 表示分组容器，false 表示叶子权限节点
»»» label	string	true	none	权限名	权限项显示名称
»»» multiple_selected	boolean	false	none	部分选中	分组节点下存在部分选中的子节点时为true
»»» translate_name	string	false	none	权限标识	权限英文标识名
»» status	integer	true	none	状态	角色状态，1 表示启用

返回示例

```
{
  "data": [
    {
      "created_by": 0,
      "description": "由超级管理员创建，可管理普通用户，不可管理其他管理员",
      "id": 2,
```

```
"is_self": true,
"is_system": true,
"label": "普通管理员",
"name": "admin_regular",
"permissions": [],
"status": 1
},
{
  "created_by": 2,
  "description": "固定角色, 权限由管理员逐项配置",
  "id": 3,
  "is_self": false,
  "is_system": false,
  "label": "gen_role_1",
  "name": "gen_role_1",
  "permissions": [
    {
      "children": [
        {
          "checked": true,
          "id": 2,
          "is_group": false,
          "label": "源列表"
        },
        {
          "checked": true,
          "children": [
            {
              "checked": true,
              "id": 4,
              "is_group": false,
              "label": "编码设置"
            }
          ],
          "id": 3,
          "is_group": false,
          "label": "预览列表"
        }
      ],
      "checked": false,
      "children": [
        {
          "checked": true,
          "id": 6,
          "is_group": false,
          "label": "分辨率帧率"
        },
        {
          "checked": true,
          "id": 7,
          "is_group": false,
          "label": "调音台"
        }
      ]
    }
  ]
}
```

```
{
  "checked": true,
  "children": [
    {
      "checked": false,
      "id": 9,
      "is_group": false,
      "label": "音频输出"
    },
    {
      "checked": false,
      "id": 10,
      "is_group": false,
      "label": "静音控制"
    },
    {
      "checked": false,
      "id": 11,
      "is_group": false,
      "label": "HDMI audio使能"
    }
  ],
  "id": 8,
  "is_group": true,
  "label": "音频输出编辑"
},
{
  "checked": false,
  "children": [
    {
      "checked": false,
      "id": 13,
      "is_group": false,
      "label": "HDMI video使能"
    }
  ],
  "id": 12,
  "is_group": true,
  "label": "视频输出编辑"
}
],
"id": 5,
"is_group": false,
"label": "物理接口输出配置"
},
{
  "checked": false,
  "children": [
    {
      "checked": false,
      "id": 15,
      "is_group": false,
      "label": "布局选择"
    }
  ]
}
```

```
    },
    {
      "checked": false,
      "children": [
        {
          "checked": false,
          "id": 17,
          "is_group": false,
          "label": "布局新增"
        },
        {
          "checked": false,
          "id": 18,
          "is_group": false,
          "label": "布局删除"
        },
        {
          "checked": false,
          "id": 19,
          "is_group": false,
          "label": "布局修改"
        },
        {
          "checked": false,
          "id": 20,
          "is_group": false,
          "label": "布局重命名"
        },
        {
          "checked": false,
          "id": 21,
          "is_group": false,
          "label": "布局保存"
        }
      ],
      "id": 16,
      "is_group": true,
      "label": "布局编辑器"
    },
    {
      "checked": false,
      "children": [
        {
          "checked": false,
          "id": 23,
          "is_group": false,
          "label": "音柱"
        },
        {
          "checked": false,
          "id": 24,
          "is_group": false,
          "label": "视频输出背景"
        }
      ]
    }
  ]
}
```

```

        },
        {
            "checked": false,
            "id": 25,
            "is_group": false,
            "label": "边框叠层"
        },
        {
            "checked": false,
            "id": 26,
            "is_group": false,
            "label": "叠层模板"
        },
        {
            "checked": false,
            "id": 27,
            "is_group": false,
            "label": "云台控制"
        },
        {
            "checked": false,
            "id": 28,
            "is_group": false,
            "label": "MPEG-TS节目"
        }
    ],
    "id": 22,
    "is_group": true,
    "label": "其它工具"
},
{
    "checked": false,
    "id": 29,
    "is_group": false,
    "label": "叠层编辑器"
}
],
"id": 14,
"is_group": false,
"label": "多画面显示配置"
}
],
"id": 1,
"is_group": true,
"label": "主页"
},
{
    "children": [
        {
            "checked": false,
            "children": [
                {
                    "checked": false,

```

```
        "id": 32,
        "is_group": false,
        "label": "流服务启停"
    },
    {
        "checked": false,
        "id": 33,
        "is_group": false,
        "label": "流服务推流源配置"
    }
],
    "id": 31,
    "is_group": false,
    "label": "流服务查看权限"
}
],
    "id": 30,
    "is_group": true,
    "label": "流服务"
},
{
    "children": [
        {
            "checked": false,
            "id": 31,
            "is_group": false,
            "label": "通用设置"
        },
        {
            "checked": false,
            "id": 32,
            "is_group": false,
            "label": "网络设置"
        },
        {
            "checked": false,
            "id": 33,
            "is_group": false,
            "label": "用户列表"
        },
        {
            "checked": false,
            "id": 38,
            "is_group": false,
            "label": "固件升级"
        },
        {
            "checked": false,
            "id": 39,
            "is_group": false,
            "label": "高级设置"
        }
    ],
    {
```

```
        "checked": false,
        "id": 40,
        "is_group": false,
        "label": "预设配置"
    },
    {
        "children": [
            {
                "checked": false,
                "id": 42,
                "is_group": false,
                "label": "恢复出厂"
            },
            {
                "checked": false,
                "id": 43,
                "is_group": false,
                "label": "设备重启"
            }
        ],
        "id": 41,
        "is_group": true,
        "label": "系统重置"
    },
    {
        "checked": false,
        "id": 44,
        "is_group": false,
        "label": "系统日志"
    },
    {
        "checked": false,
        "id": 45,
        "is_group": false,
        "label": "角色权限管理"
    }
],
    "id": 34,
    "is_group": true,
    "label": "设置"
}
],
"status": 1
},
{
    "created_by": 2,
    "description": "固定角色, 权限由管理员逐项配置",
    "id": 4,
    "is_self": false,
    "is_system": false,
    "label": "通用角色2",
    "name": "gen_role_2",
    "permissions": [
```

```
{
  "children": [
    {
      "checked": false,
      "id": 2,
      "is_group": false,
      "label": "源列表"
    },
    {
      "checked": false,
      "children": [
        {
          "checked": false,
          "id": 4,
          "is_group": false,
          "label": "编码设置"
        }
      ],
      "id": 3,
      "is_group": false,
      "label": "预览列表"
    },
    {
      "checked": false,
      "children": [
        {
          "checked": false,
          "id": 6,
          "is_group": false,
          "label": "分辨率帧率"
        },
        {
          "checked": false,
          "id": 7,
          "is_group": false,
          "label": "调音台"
        },
        {
          "checked": false,
          "children": [
            {
              "checked": false,
              "id": 9,
              "is_group": false,
              "label": "音频输出"
            },
            {
              "checked": false,
              "id": 10,
              "is_group": false,
              "label": "静音控制"
            }
          ],
          "id": 8,
          "is_group": true,
          "label": "其他设置"
        }
      ],
      "id": 5,
      "is_group": true,
      "label": "高级设置"
    }
  ],
  "id": 1,
  "is_group": true,
  "label": "设置"
}
```

```
        "checked": false,
        "id": 11,
        "is_group": false,
        "label": "HDMI audio使能"
    }
],
    "id": 8,
    "is_group": true,
    "label": "音频输出编辑"
},
{
    "checked": false,
    "children": [
        {
            "checked": false,
            "id": 13,
            "is_group": false,
            "label": "HDMI video使能"
        }
    ],
    "id": 12,
    "is_group": true,
    "label": "视频输出编辑"
}
],
    "id": 5,
    "is_group": false,
    "label": "物理接口输出配置"
},
{
    "checked": false,
    "children": [
        {
            "checked": false,
            "id": 15,
            "is_group": false,
            "label": "布局选择"
        }
    ],
    {
        "checked": false,
        "children": [
            {
                "checked": false,
                "id": 17,
                "is_group": false,
                "label": "布局新增"
            }
        ],
        {
            "checked": false,
            "id": 18,
            "is_group": false,
            "label": "布局删除"
        }
    ],
}
```

```
{
  {
    "checked": false,
    "id": 19,
    "is_group": false,
    "label": "布局修改"
  },
  {
    "checked": false,
    "id": 20,
    "is_group": false,
    "label": "布局重命名"
  },
  {
    "checked": false,
    "id": 21,
    "is_group": false,
    "label": "布局保存"
  }
],
"id": 16,
"is_group": true,
"label": "布局编辑器"
},
{
  "checked": false,
  "children": [
    {
      "checked": false,
      "id": 23,
      "is_group": false,
      "label": "音柱"
    },
    {
      "checked": false,
      "id": 24,
      "is_group": false,
      "label": "视频输出背景"
    },
    {
      "checked": false,
      "id": 25,
      "is_group": false,
      "label": "边框叠层"
    },
    {
      "checked": false,
      "id": 26,
      "is_group": false,
      "label": "叠层模板"
    },
    {
      "checked": false,
      "id": 27,
```

```
        "is_group": false,
        "label": "云台控制"
    },
    {
        "checked": false,
        "id": 28,
        "is_group": false,
        "label": "MPEG-TS节目"
    }
],
"id": 22,
"is_group": true,
"label": "其它工具"
},
{
    "checked": false,
    "id": 29,
    "is_group": false,
    "label": "叠层编辑器"
}
],
"id": 14,
"is_group": false,
"label": "多画面显示配置"
}
],
"id": 1,
"is_group": true,
"label": "主页"
},
{
    "children": [
        {
            "checked": false,
            "children": [
                {
                    "checked": false,
                    "id": 32,
                    "is_group": false,
                    "label": "流服务启停"
                },
                {
                    "checked": false,
                    "id": 33,
                    "is_group": false,
                    "label": "流服务推流源配置"
                }
            ],
            "id": 31,
            "is_group": false,
            "label": "流服务查看权限"
        }
    ]
},
```

```
"id": 30,
"is_group": true,
"label": "流服务"
},
{
  "children": [
    {
      "checked": false,
      "id": 31,
      "is_group": false,
      "label": "通用设置"
    },
    {
      "checked": false,
      "id": 32,
      "is_group": false,
      "label": "网络设置"
    },
    {
      "checked": false,
      "id": 33,
      "is_group": false,
      "label": "用户列表"
    },
    {
      "checked": false,
      "id": 38,
      "is_group": false,
      "label": "固件升级"
    },
    {
      "checked": false,
      "id": 39,
      "is_group": false,
      "label": "高级设置"
    },
    {
      "checked": false,
      "id": 40,
      "is_group": false,
      "label": "预设配置"
    },
    {
      "children": [
        {
          "checked": false,
          "id": 42,
          "is_group": false,
          "label": "恢复出厂"
        },
        {
          "checked": false,
          "id": 43,
```

```
        "is_group": false,
        "label": "设备重启"
    }
],
    "id": 41,
    "is_group": true,
    "label": "系统重置"
},
{
    "checked": false,
    "id": 44,
    "is_group": false,
    "label": "系统日志"
},
{
    "checked": false,
    "id": 45,
    "is_group": false,
    "label": "角色权限管理"
}
],
    "id": 34,
    "is_group": true,
    "label": "设置"
}
],
    "status": 1
},
{
    "created_by": 2,
    "description": "固定角色，权限由管理员逐项配置",
    "id": 5,
    "is_self": false,
    "is_system": false,
    "label": "通用角色3",
    "name": "gen_role_3",
    "permissions": [
        {
            "children": [
                {
                    "checked": false,
                    "id": 2,
                    "is_group": false,
                    "label": "源列表"
                },
                {
                    "checked": false,
                    "children": [
                        {
                            "checked": false,
                            "id": 4,
                            "is_group": false,
                            "label": "编码设置"
                        }
                    ]
                }
            ]
        }
    ]
}
```

```

    }
  ],
  "id": 3,
  "is_group": false,
  "label": "预览列表"
},
{
  "checked": false,
  "children": [
    {
      "checked": false,
      "id": 6,
      "is_group": false,
      "label": "分辨率帧率"
    },
    {
      "checked": false,
      "id": 7,
      "is_group": false,
      "label": "调音台"
    },
    {
      "checked": false,
      "children": [
        {
          "checked": false,
          "id": 9,
          "is_group": false,
          "label": "音频输出"
        },
        {
          "checked": false,
          "id": 10,
          "is_group": false,
          "label": "静音控制"
        },
        {
          "checked": false,
          "id": 11,
          "is_group": false,
          "label": "HDMI audio使能"
        }
      ]
    },
    {
      "id": 8,
      "is_group": true,
      "label": "音频输出编辑"
    }
  ],
  {
    "checked": false,
    "children": [
      {
        "checked": false,
        "id": 13,

```

```
        "is_group": false,
        "label": "HDMI video使能"
    }
],
    "id": 12,
    "is_group": true,
    "label": "视频输出编辑"
}
],
    "id": 5,
    "is_group": false,
    "label": "物理接口输出配置"
},
{
    "checked": false,
    "children": [
        {
            "checked": false,
            "id": 15,
            "is_group": false,
            "label": "布局选择"
        },
        {
            "checked": false,
            "children": [
                {
                    "checked": false,
                    "id": 17,
                    "is_group": false,
                    "label": "布局新增"
                },
                {
                    "checked": false,
                    "id": 18,
                    "is_group": false,
                    "label": "布局删除"
                },
                {
                    "checked": false,
                    "id": 19,
                    "is_group": false,
                    "label": "布局修改"
                },
                {
                    "checked": false,
                    "id": 20,
                    "is_group": false,
                    "label": "布局重命名"
                },
                {
                    "checked": false,
                    "id": 21,
                    "is_group": false,
```

```
        "label": "布局保存"
    },
    ],
    "id": 16,
    "is_group": true,
    "label": "布局编辑器"
},
{
    "checked": false,
    "children": [
        {
            "checked": false,
            "id": 23,
            "is_group": false,
            "label": "音柱"
        },
        {
            "checked": false,
            "id": 24,
            "is_group": false,
            "label": "视频输出背景"
        },
        {
            "checked": false,
            "id": 25,
            "is_group": false,
            "label": "边框叠层"
        },
        {
            "checked": false,
            "id": 26,
            "is_group": false,
            "label": "叠层模板"
        },
        {
            "checked": false,
            "id": 27,
            "is_group": false,
            "label": "云台控制"
        },
        {
            "checked": false,
            "id": 28,
            "is_group": false,
            "label": "MPEG-TS节目"
        }
    ],
    "id": 22,
    "is_group": true,
    "label": "其它工具"
},
{
    "checked": false,
```

```
        "id": 29,
        "is_group": false,
        "label": "叠层编辑器"
    }
],
    "id": 14,
    "is_group": false,
    "label": "多画面显示配置"
}
],
    "id": 1,
    "is_group": true,
    "label": "主页"
},
{
    "children": [
        {
            "checked": false,
            "children": [
                {
                    "checked": false,
                    "id": 32,
                    "is_group": false,
                    "label": "流服务启停"
                },
                {
                    "checked": false,
                    "id": 33,
                    "is_group": false,
                    "label": "流服务推流源配置"
                }
            ],
            "id": 31,
            "is_group": false,
            "label": "流服务查看权限"
        }
    ],
    "id": 30,
    "is_group": true,
    "label": "流服务"
},
{
    "children": [
        {
            "checked": false,
            "id": 31,
            "is_group": false,
            "label": "通用设置"
        },
        {
            "checked": false,
            "id": 32,
            "is_group": false,
```

```
    "label": "网络设置"
  },
  {
    "checked": false,
    "id": 33,
    "is_group": false,
    "label": "用户列表"
  },
  {
    "checked": false,
    "id": 38,
    "is_group": false,
    "label": "固件升级"
  },
  {
    "checked": false,
    "id": 39,
    "is_group": false,
    "label": "高级设置"
  },
  {
    "checked": false,
    "id": 40,
    "is_group": false,
    "label": "预设配置"
  },
  {
    "children": [
      {
        "checked": false,
        "id": 42,
        "is_group": false,
        "label": "恢复出厂"
      },
      {
        "checked": false,
        "id": 43,
        "is_group": false,
        "label": "设备重启"
      }
    ],
    "id": 41,
    "is_group": true,
    "label": "系统重置"
  },
  {
    "checked": false,
    "id": 44,
    "is_group": false,
    "label": "系统日志"
  },
  {
    "checked": false,
```

```

        "id": 45,
        "is_group": false,
        "label": "角色权限管理"
      }
    ],
    "id": 34,
    "is_group": true,
    "label": "设置"
  }
],
"status": 1
}
],
"result": "ok"
}

```

3.7 POST /roles/permupdate

功能说明

更新角色信息与权限。通过 `role_id` 指定目标角色，可同时更新角色名称、描述及权限节点 ID 列表。`name`、`description`、`permission_ids` 均为可选字段，仅提交需更新的内容。预设系统角色（admin、admin_regular）不可修改权限。仅管理员可执行此操作。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 <code>{"token": "xxx"}</code>
body	body	object	否	请求体	包含角色更新信息的 JSON 对象
» role_id	body	integer	是	角色ID	要修改权限的角色ID
» name	body	string	否	角色名	角色标识名称，不传则不更新
» description	body	string	否	描述	角色描述信息，不传则不更新
» permission_ids	body	[integer]	否	权限ID列表	该角色拥有的权限节点ID列表，不传则不更新

Body 请求参数

```

{
  "role_id": 3,
  "name": "gen_role_1",
  "description": "自定义描述",
  "permission_ids": [2, 3, 4, 5, 6, 7, 8]
}

```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none	返回结果	固定返回 "ok" 表示角色权限更新成功

返回示例

```
{
  "result": "ok"
}
```

3.8 POST /users/logout

功能说明

退出登录接口。调用后使当前会话令牌立即失效，后续所有使用该令牌请求将被拒绝。客户端需重新调用登录接口获取新令牌才能继续访问其他 API。

注意：注销的 token 从请求头 app（或 Cookie）中获取，**不从 body 读取**。仅在 body 里放 token 不会生效，请与其他接口一致地在 app 头携带 token。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 {"token": "xxx"}。服务端从此头部提取 token 进行注销

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	none	返回结果	固定返回 "ok" 表示注销成功
» data	object	false	none	返回数据	注销成功后返回空对象

返回示例

```
{
  "result": "ok",
  "data": {}
}
```

3.9 POST /users/auth_info_get

功能说明

获取鉴权配置信息。返回当前系统是否启用 Token 鉴权机制（enable），以及当前登录用户是否有权限修改鉴权开关（canModify，仅管理员为 true）。可用于前端判断是否展示鉴权设置选项。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 {"token": "xxx"}

返回数据结构

名称	类型	必选	约束	中文名	说明
» result	string	true	none	返回结果	固定返回 "ok" 表示请求成功
» data	object	true	none	返回数据	鉴权配置信息对象
»» enable	boolean	true	none	鉴权启用	是否已开启 Token 鉴权机制
»» canModify	boolean	true	none	可修改	当前用户是否为管理员，即是否有权限修改鉴权开关

状态码 200

返回示例

```
{
  "result": "ok",
  "data": {
    "enable": true,
    "canModify": true
  }
}
```

3.10 POST /users/auth_info_set

功能说明

设置鉴权开关。管理员可启用或禁用系统的 Token 鉴权机制。禁用后所有 API 请求不再校验 app 头中的 token，但强烈建议生产环境保持启用以确保设备安全。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 <code>{"token": "xxx"}</code>
enable	body	boolean	是	启用鉴权	是否启用 Token 鉴权，必须是布尔类型 (true/false)

Body 请求参数

```
{
  "enable": false
}
```

返回数据结构

名称	类型	必选	约束	中文名	说明
» result	string	true	none	返回结果	固定返回 <code>"ok"</code> 表示设置成功
» data	object	true	none	返回数据	设置结果对象
»» enable	boolean	true	none	启用状态	修改后的鉴权启用状态

状态码 200

返回示例

```
{
  "result": "ok",
  "data": {
    "enable": false
  }
}
```

3.11 GET /users/info

功能说明

根据用户名查询指定用户的详细信息，返回包括用户 ID、别名、角色、创建时间、密码（明文）、联系电话、描述等完整字段。

重要提示：

虽然该接口标注为 GET 方法，但查询参数（`username`）需通过请求体（body）以 JSON 格式传递，而非 URL 查询参数。这是非标准 REST 风格的实现，集成时请注意将 `username` 放在 JSON 请求体中发送。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 <code>{"token": "xxx"}</code>
body	body	object	否	请求体	包含查询条件的 JSON 对象（注意：GET 方法使用 body 传参）
» username	body	string	是	用户名	要查询的用户登录用户名

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
result	string	true	none	返回结果	固定返回 <code>"ok"</code> 表示查询成功
» data	object	true	none	返回数据	用户的完整信息（参见 UserEntity 全部字段）

返回示例

```
{
  "data": {
    "alias": "test1",
    "changed": true,
    "create_time": "169733754",
    "created_by": 1,
    "descript": "",
    "id": 2,
    "password": "admin1",
    "phone": "",
    "role_id": 2,
    "status": 1,
    "token": "9b8b92fb007a86dee14f5d89bf0c4d2a",
  }
}
```

```
    "updated_at": "1970-01-03T07:08:53",
    "username": "admin1"
  },
  "result": "ok"
}
```

3.12 POST /users/session/check

功能说明

校验当前会话 token 是否有效。将 token 放在 `app` 请求头中发送，服务端验证后返回对应用户的基础信息（用户名和别名）。常用于客户端启动时检查本地缓存的 token 是否仍有效，避免不必要的重新登录流程。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 <code>{"token":"xxx"}</code> 。服务端将校验此令牌的有效性

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
result	string	true	none	返回结果	校验通过返回 <code>"ok"</code> ，校验失败返回 <code>"error"</code>
» data	object	false	none	返回数据	校验通过时返回用户基础信息；校验失败时不返回此字段
»» alias	string	true	none	别名	用户的别名
»» username	string	true	none	用户名	用户的登录用户名

返回示例

校验通过

```
{
  "result": "ok",
  "data": {
    "alias": "admin",
    "username": "admin"
  }
}
```

```
{
  "result": "error",
  "msg": "300"
}
```

3.13 POST /auth/check

功能说明

验证当前用户的身份认证信息，并返回叠层编辑器跨进程权限校验结果（perm_ck）。该接口主要用于叠层编辑器（overlayprocess）子进程校验当前操作用户是否有相应的操作权限。当请求触发叠层权限路径检查时，响应中会额外包含 perm_ck 对象。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌，格式为 JSON 字符串 {"token": "xxx"}
body	body	object	否	请求体	包含 overlayprocess 代理的 method 和 url 信息

返回数据结构

名称	类型	必选	约束	中文名	说明
result	string	true	none	返回结果	固定返回 "ok" 表示认证通过
perm_ck	object	false	none	权限校验结果	仅在触发叠层权限检查路径时返回此字段
» result	string	true	none	返回结果	权限校验结果，"ok" 表示通过，"error" 表示无权限
» msg	string	false	none	错误码	权限不足时返回的错误码字符串
» permission_id	integer	false	none	权限ID	被拒绝的权限项 ID
» translate_name	string	false	none	权限标识	被拒绝的权限英文标识名

状态码 200

返回示例

```
{
  "result": "ok",
  "perm_ck": {
    "result": "ok"
  }
}
```

4. OVERLAY模块

OVERLAY（叠加层）模块用于管理视频输出上的叠加模板（OSD 叠加层）。每个叠加模板可包含图片和文字元素，支持元素的添加、删除、位置/大小/层级调整等编辑操作，以及模板的保存、预览、复制、导出功能。叠加层常用于在视频画面上叠加台标、时钟、日期、文本等辅助信息。

4.1 GET /overlay/list

功能说明

获取所有叠加模板的列表，返回模板的基本信息及当前编辑/选中状态。

请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	否	输出ID	按指定输出通道筛选叠加模板
app	header	string	是	应用标识	请求来源应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	模板列表	叠加模板数组
»» background	string	false	无	背景颜色	模板背景颜色，如 "transparent"、"output1"、"output2"
»» editing	boolean	false	无	编辑状态	该模板是否正在被编辑
»» id	integer	false	无	模板ID	模板唯一标识符
»» img_path	string	false	无	预览图路径	模板预览图片的本地路径
»» modified	boolean	false	无	已修改	模板自上次保存后是否有未保存的修改
»» name	string	false	无	模板名称	用户自定义的模板名称

名称	类型	必选	约束	中文名	说明
»» selected	boolean	false	无	选中状态	该模板是否被选中（勾选）用于输出叠加
»» status	integer	false	无	状态	模板当前状态，0 表示正常
» result	string	true	无	操作结果	请求结果，"ok" 表示成功

返回示例

```
{
  "data": [
    {
      "background": "transparent",
      "editing": true,
      "id": 1,
      "img_path": "./data/decoder_config/overlay/preview/overlay_1.png",
      "modified": false,
      "name": "template_1",
      "selected": false,
      "status": 0
    }
  ],
  "result": "ok"
}
```

4.2 POST /overlay/add

功能说明

在叠加模板列表中新增一个空白模板。新模板的默认名称为 "new_template"，默认背景为 "transparent"。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果，"ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.3 POST /overlay/remove

功能说明

删除指定的叠加模板。删除操作不可逆，已绑定到输出的模板无法删除。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	body	integer	是	叠加模板ID	待删除的叠加模板唯一标识
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "overlay_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.4 POST /overlay/copy

功能说明

复制指定的叠加模板，生成一个包含相同元素内容的新模板。新模板名称为原名称后追加 "_copy"。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	body	integer	是	源模板ID	待复制的叠加模板标识
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "overlay_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.5 POST /overlay/rename

功能说明

重命名指定的叠加模板。

请求参数

名称	位置	类型	必选	中文名	说明
id	body	integer	是	模板ID	待重命名的模板标识
name	body	string	是	新名称	模板的新名称字符串
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "id": 1,
  "name": "template_1"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.6 POST /overlay/save

功能说明

持久化保存当前叠加模板的所有修改（包括元素的位置、属性变更等）。编辑中的模板在保存前修改不会生效到输出画面。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	body	integer	是	叠加模板ID	待保存的叠加模板标识
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "overlay_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.7 POST /overlay/cancel

功能说明

撤销当前编辑中叠加模板的所有未保存修改，恢复至上次保存的状态。仅影响元素的位置和属性，不删除模板本身。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	body	integer	是	叠加模板ID	待取消编辑的叠加模板标识
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "overlay_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.8 POST /overlay/select

功能说明

选择（勾选）某一叠加模板并将其绑定到指定的输出通道。被选中的模板将在对应输出通道的视频画面上叠加显示。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» overlay_id	body	integer	是	叠加模板ID	待选择的叠加模板标识
» output_id	body	integer	是	输出ID	目标输出通道标识

Body 请求参数

```
{
  "overlay_id": 1,
  "output_id": 0
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

4.9 POST /overlay/items/set

功能说明

设置正在编辑的叠加模板中的图片和文字元素的全部属性。支持全量替换模板内所有元素（包括顺序、位置、样式等）。每次调用应传入完整的 list 数组。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	body	integer	是	叠加模板 ID	目标叠加模板的唯一标识
mode	body	string	否	编辑模式	操作模式，如 "z_index" 表示仅调整层级
list	body	[object]	是	元素列表	模板中包含的所有元素数组
» id	body	integer	否	元素 ID	元素的唯一标识，新建元素可省略
» img_id	body	string null	否	图片 ID	元素关联的图片ID，仅当 type="image" 时有效
» width	body	integer	否	宽度	元素宽度，以百分比表示（1-1000，1000=100%）
» height	body	integer	否	高度	元素高度，以百分比表示（1-1000，1000=100%）
» x	body	integer	否	水平位置	元素左上角 X 坐标，以百分比表示
» y	body	integer	否	垂直位置	元素左上角 Y 坐标，以百分比表示
» z_index	body	integer	否	层级	元素的 Z 轴层级，数值越大显示越靠前
» type	body	string	否	元素类型	叠加类型，"image" 或 "text"
» opacity	body	number	否	不透明度	元素透明度，范围 0.0（完全透明）~1.0（完全不透明）

名称	位置	类型	必选	中文名	说明
» text_type	body	string	否	文本类型	文本来源类型, "custom" (自定义)、"time" (时间)、"date" (日期)
» font_color	body	string	否	字体颜色	文本颜色, 格式为 "0xRRGGBB", 如 "0xffffffff"
» stroke_enable	body	boolean	否	描边开关	是否启用文本描边效果
» stroke_color	body	string	否	描边颜色	描边颜色, 格式同 font_color
» stroke_size	body	integer	否	描边宽度	描边线条宽度, 单位像素
» content	body	string	否	文本内容	自定义文本内容, 仅 text_type="custom" 时有效
» format	body	string	否	日期格式	时间/日期格式化字符串, 仅 text_type 为 "time" 或 "date" 时有效
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "overlay_id": 1,
  "mode": "z_index",
  "list": [
    {
      "id": 1,
      "img_id": 1,
      "img_path": "/data/decoder_config/overlay/img/1.png",
      "width": 1000,
      "height": 500,
      "x": 10,
      "y": 0,
      "z_index": 0,
      "type": "image",
      "opacity": 1.0
    },
    {
      "id": 2,
      "x": 0,
      "y": 0,
      "z_index": 0,
      "width": 100,
      "height": 100,
      "text_type": "",

```

```
        "font_color": "0xffffffff",
        "opacity": 0,
        "stroke_enable": false,
        "stroke_color": "0xffffffff",
        "stroke_size": 18,
        "content": "",
        "type": "text",
        "format": ""
    }
]
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.10 GET /overlay/items/get

功能说明

获取指定叠加模板的全部元素信息，包括模板基本信息、图片元素和文字元素的完整属性列表。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	query	string	否	叠加模板ID	目标叠加模板的唯一标识
app	header	string	是	应用标识	请求来源应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	无	模板详情	单个模板的完整信息对象
»» background	string	false	无	背景颜色	模板背景, 可为 "transparent"、"output1"、"output2"
»» name	string	false	无	模板名称	模板名称字符串

名称	类型	必选	约束	中文名	说明
»» id	integer	false	无	模板ID	模板唯一标识
»» list	[object]	true	无	元素列表	模板包含的所有元素数组
»»» id	integer	false	无	元素ID	组件的唯一标识
»»» img_url	string	false	无	图片URL	上传图片的访问 URL
»»» img_id	integer	false	无	图片ID	上传图片的唯一标识
»»» width	integer	false	无	宽度	元素宽度，百分比值
»»» height	integer	false	无	高度	元素高度，百分比值
»»» x	integer	false	无	X坐标	元素左上角 X 坐标，百分比值
»»» y	integer	false	无	Y坐标	元素左上角 Y 坐标，百分比值
»»» z_index	integer	false	无	层级	元素 Z 轴层级，数值越大越靠前
»»» type	string	false	无	元素类型	组件类型: "image" 或 "text"
»»» opacity	number	false	无	不透明度	元素透明度，范围 0.0~1.0
»»» path	string	false	无	图片路径	图片本地存储路径
»»» text_type	string	false	无	文本类型	文本来源: "custom"、"time"、"date"
»»» font_color	string	false	无	字体颜色	文本颜色，格式 "0xRRGGBB"
»»» stroke_enable	boolean	false	无	描边开关	文本描边是否启用
»»» stroke_color	string	false	无	描边颜色	描边颜色，格式同 font_color
»»» stroke_size	integer	false	无	描边宽度	描边线条宽度，单位像素
»»» content	string	false	无	文本内容	自定义文本的实际内容
»»» format	string	false	无	日期格式	时间/日期格式化字符串
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok",
  "data": {
    "id": 1,
    "name": "overlay1",
    "background": "transparent",
    "list": [
      {
        "id": 1,
        "width": 100,
        "height": 100,
        "x": 10,
        "y": 0,
        "z_index": 0,
        "type": "image",
        "opacity": 0,
        "img_id": 2
      },
      {
        "id": 2,
        "x": 0,
        "y": 0,
        "z_index": 0,
        "width": 100,
        "height": 100,
        "text_type": "custom",
        "font_color": "0xffffffff",
        "opacity": 0,
        "stroke_enable": false,
        "stroke_color": "0xffffffff",
        "stroke_size": 18,
        "content": "",
        "type": "text"
      },
      {
        "id": 3,
        "x": 0,
        "y": 0,
        "z_index": 0,
        "width": 100,
        "height": 100,
        "text_type": "time",
        "format": "hh:mm:ss",
        "font_color": "0xffffffff",
        "opacity": 0,
        "stroke_enable": false,
        "stroke_color": "0xffffffff",
        "stroke_size": 18,
        "content": "",
        "type": "text"
      }
    ]
  }
}
```

```
{
  "id": 4,
  "x": 0,
  "y": 0,
  "z_index": 0,
  "width": 100,
  "height": 100,
  "text_type": "date",
  "format": "yyyy/MM/dd",
  "font_color": "0xffffffff",
  "opacity": 0,
  "stroke_enable": false,
  "stroke_color": "0xffffffff",
  "stroke_size": 18,
  "content": "",
  "type": "text"
}
]
```

说明：`background` 字段可取值为 "transparent"、"output1"、"output2"，分别表示透明背景、输出1画面为背景、输出2画面为背景。

4.11 POST /overlay/items/add

功能说明

向指定叠加模板中添加一个新的图片或文本元素。成功添加后会返回该元素的 ID。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	body	integer	是	叠加模板ID	目标叠加模板的唯一标识
type	body	string	是	元素类型	元素类型："image" 或 "text"
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "overlay_id": 1,
  "type": "image"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.12 POST /overlay/items/remove

功能说明

删除指定叠加模板中的某个图片或文本元素。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	body	integer	是	叠加模板ID	目标叠加模板的唯一标识
item_id	body	integer	是	元素ID	待删除的图片或文本元素ID
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "overlay_id": 1,
  "item_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.13 GET /overlay/images/list

功能说明

获取已上传至设备的图片资源列表，包含图片的 ID、本地路径和尺寸信息。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	图片列表	图片资源数组
»» id	integer	false	无	图片ID	图片的唯一标识
»» img_path	string	false	无	图片路径	图片本地存储路径
»» width	integer	false	无	宽度	图片原始宽度，单位像素
»» height	integer	false	无	高度	图片原始高度，单位像素
» result	string	true	无	操作结果	请求结果，"ok" 表示成功

返回示例

```
{
  "data": [
    {
      "id": 1,
      "img_path": "./test1.png",
      "width": 1920,
      "height": 1080
    },
    {
      "id": 2,
      "img_path": "./test2.png",
      "width": 1920,
      "height": 1080
    }
  ],
  "result": "ok"
}
```

4.14 POST /overlay/images/remove

功能说明

从设备中删除指定 ID 的已上传图片资源。

请求参数

名称	位置	类型	必选	中文名	说明
img_id	body	integer	是	图片ID	待删除图片的唯一标识

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "img_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.15 POST /overlay/images/add

功能说明

上传一张图片文件到设备，供叠加模板使用。请求体应为 multipart/form-data 格式，使用 `img_url` 字段携带文件。

请求参数

名称	位置	类型	必选	中文名	说明
img_url	body	file	是	图片文件	待上传的图片文件
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

此接口使用 multipart/form-data 格式，文件字段名为 `img_url`。

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.16 GET /overlay/editing/get

功能说明

获取当前正在编辑中的叠加模板信息及其预览图片路径。编辑模式用于在编辑器中实时预览模板效果。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	body	integer	是	编辑模板ID	正在编辑中的叠加模板ID
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "overlay_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功
» data	object	true	无	编辑模板信息	编辑中的模板详情
»» overlay_id	integer	false	无	编辑模板ID	当前编辑中的叠加模板ID
»» editing_path	string	false	无	编辑路径	编辑预览图的本地路径

返回示例

```
{
  "data": {
    "overlay_id": 1,
    "editing_path": ""
  },
  "result": "ok"
}
```

4.17 POST /overlay/editing/select

功能说明

设置指定叠加模板为当前编辑中的模板。同一时间只能有一个模板处于编辑状态。

请求参数

名称	位置	类型	必选	中文名	说明
overlay_id	body	integer	是	编辑模板ID	指定进入编辑模式的叠加模板ID
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "overlay_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{"result": "ok"}
```

4.18 GET /overlay/preview/get

功能说明

获取已选中并叠加到指定输出通道视频上的模板预览图片路径。

请求参数

名称	位置	类型	必选	中文名	说明
output_id	body	integer	是	输出ID	目标输出通道标识
app	header	string	是	应用标识	请求来源应用标识

Body 请求参数

```
{
  "output_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功
» data	object	true	无	预览信息	预览模板详情
»» overlay_id	integer	false	无	模板ID	当前输出上叠加的模板ID
»»» preview_path	string	false	无	预览路径	叠加预览图的本地路径

返回示例

```
{
  "data": {
    "overlay_id": 1,
    "preview_path": ""
  },
  "result": "ok"
}
```

4.19 GET /overlay/datetime/format/list

功能说明

获取叠加文本支持的日期和时间格式化选项列表。每种格式给出一个 label（示例展示）和对应的 format 字符串。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功
» data	object	true	无	格式选项	日期和时间格式分组
»» date	[object]	true	无	日期格式	日期格式化选项列表
»»» id	integer	false	无	格式ID	日期格式唯一标识
»»» label	string	false	无	日期示例	格式对应的显示示例
»»» format	string	false	无	日期格式	日期格式化字符串

名称	类型	必选	约束	中文名	说明
»» time	[object]	true	无	时间格式	时间格式化选项列表
»»» id	integer	false	无	格式ID	时间格式唯一标识
»»» label	string	false	无	时间示例	格式对应的显示示例
»»» format	string	false	无	时间格式	时间格式化字符串

返回示例

```
{
  "data": {
    "date": [
      {
        "id": 1,
        "label": "*3/14/2012",
        "format": "*M/dd/yyyy"
      },
      {
        "id": 2,
        "label": "*March 14, 2012",
        "format": "*MMMM dd, yyyy"
      },
      {
        "id": 3,
        "label": "03-14-2012",
        "format": "MM-dd-yyyy"
      },
      {
        "id": 4,
        "label": "March 14, 2012",
        "format": "MMMMM dd, yyyy"
      },
      {
        "id": 5,
        "label": "*March 14, 2012",
        "format": "*MMMM dd, yyyy"
      },
      {
        "id": 6,
        "label": "March 2012",
        "format": "MMMMM yyyy"
      },
      {
        "id": 7,
        "label": "March 14",
        "format": "MMMM dd"
      }
    ],
    "time": [
      {
```

```
    "id": 1,
    "label": "*1:30:55 PM",
    "format": "*h:mm:ss A"
  },
  {
    "id": 2,
    "label": "1:30 PM",
    "format": "h:mm A"
  },
  {
    "id": 3,
    "label": "1:30:55 PM",
    "format": "h:mm:ss A"
  },
  {
    "id": 4,
    "label": "3/14/12 1:30 PM",
    "format": "M/dd/yy h:mm A"
  },
  {
    "id": 5,
    "label": "3/14/12 1:30:55 PM",
    "format": "M/dd/yy h:mm:ss A"
  }
]
},
"result": "ok"
}
```

4.20 GET /overlay/font/list

功能说明

获取叠加文本支持的字体列表。返回字体 ID、显示标签和字体名称 (value) 供前端选择。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功
» data	[object]	true	无	字体列表	字体选项数组
»» id	integer	false	无	字体ID	字体唯一标识

名称	类型	必选	约束	中文名	说明
»» label	string	false	无	字体标签	字体显示名称
»» value	string	false	无	字体值	字体文件名称

返回示例

```
{
  "data": [
    {
      "id": 1,
      "label": "",
      "value": ""
    },
    {
      "id": 2,
      "label": "",
      "value": ""
    }
  ],
  "result": "ok"
}
```

5. SOURCE模块

SOURCE（视频源）模块用于管理视频源分组（组）及各组下的视频流（流）。视频源按「组」组织，组内可添加多种协议（RTSP / RTMP / RTMPS / SRT / RTP / UDP / HTTP / HTTPS / NDI / ZIXI / RTMP_SERVER 等）的拉流配置，供布局窗格与输出调用。

5.1 GET /source/groups/list

功能说明

获取所有视频源分组及组内流信息。支持通过 `is_need_stream` 参数控制是否一并返回各组的流列表。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» is_need_stream	body	boolean	否	是否携带流信息	设为 true 时返回组内流列表，false 时仅返回组基本信息

Body 请求参数

```
{
  "is_need_stream": true
}
```

返回数据结构

状态码 200

公共字段（所有流类型均返回）

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	分组列表	视频源分组数组
»» id	string	true	无	分组ID	组的唯一标识（MD5 哈希值）
»» name	string	true	无	组名称	组的显示名称
»» streams	[object]	true	无	流列表	组内视频流数组（is_need_stream=true 时返回）

以下为 `streams[]` 中各协议类型专属字段。所有流对象均包含 `id`、`name`、`type`、`url`、`status`、`buffer`、`enable`、`preview_mode` 等公共基础字段。

RTSP 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识（MD5 哈希值）
»»» name	string	true	无	流名称	流的显示名称
»»» type	string	true	无	协议类型	流协议类型（rtsp）
»»» url	string	true	无	流地址	RTSP 拉流地址
»»» status	integer	true	无	状态	拉流状态：0=未连接，1=正在连接，2=连接成功，3=连接失败，4=性能受限停止
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式：Auto、Full、Preview
»»» trans_mode	string	true	无	传输模式	RTSP 底层传输协议："tcp" 或 "udp"

名称	类型	必选	约束	中文名	说明
»»» connect_speed	integer	true	无	连接超时	连接超时时间（ms）
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值（ms）
»»» user	string	true	无	鉴权账号	RTSP 鉴权用户名
»»» password	string	true	无	鉴权密码	RTSP 鉴权密码

RTMP 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识
»»» name	string	true	无	流名称	流的显示名称
»»» type	string	true	无	协议类型	流协议类型（rtmp）
»»» url	string	true	无	流地址	RTMP 拉流地址
»»» status	integer	true	无	状态	拉流状态
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式
»»» connect_speed	integer	true	无	连接超时	连接超时时间（ms）
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值（ms）
»»» user	string	true	无	鉴权账号	RTMP 鉴权用户名
»»» password	string	true	无	鉴权密码	RTMP 鉴权密码

RTMPS 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识
»»» name	string	true	无	流名称	流的显示名称
»»» type	string	true	无	协议类型	流协议类型（rtmps）

名称	类型	必选	约束	中文名	说明
»»» url	string	true	无	流地址	RTMPS 拉流地址
»»» status	integer	true	无	状态	拉流状态
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式
»»» connect_speed	integer	true	无	连接超时	连接超时时间（ms）
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值（ms）
»»» user	string	true	无	鉴权账号	RTMPS 鉴权用户名
»»» password	string	true	无	鉴权密码	RTMPS 鉴权密码

SRT 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识
»»» name	string	true	无	流名称	流的显示名称
»»» type	string	true	无	协议类型	流协议类型（srt）
»»» url	string	true	无	流地址	SRT 拉流地址
»»» status	integer	true	无	状态	拉流状态
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式
»»» connection_mode	string	true	无	连接模式	Listener、Caller 或 Rendezvous
»»» latency	integer	true	无	延迟	SRT 数据包传输延迟（ms），-1 表示未设置
»»» connect_speed	integer	true	无	连接超时	连接超时时间（ms）
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值（ms）
»»» address	string	true	无	对端IP	对端 IP 地址
»»» listener_port	integer	true	无	监听端口	监听端口号，-1 表示未设置

名称	类型	必选	约束	中文名	说明
»»» srt_id	string	true	无	SRT流ID	SRT 自定义流标识
»»» encryption	string	true	无	加密方式	"none"、"AES-128"、"AES-192" 或 "AES-256"
»»» passphrase	string	true	无	密钥	加密密钥字符串
»»» bandwidth	integer	true	无	带宽开销	恢复带宽开销百分比，-1 表示未设置
»»» payload_size	integer	true	无	载荷大小	SRT 载荷大小（字节），-1 表示未设置
»»» show_advanced	string	true	无	显示高级	是否显示高级设置

RTMP Server 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识
»»» name	string	true	无	流名称	流的显示名称
»»» type	string	true	无	协议类型	流协议类型（rtmp_server）
»»» url	string	true	无	流地址	RTMP Server 推流地址
»»» status	integer	true	无	状态	拉流状态
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式
»»» address	string	true	无	公网IP	公网 IP 地址
»»» application	string	true	无	应用名称	RTMP 应用名称
»»» session	string	true	无	会话名称	RTMP 会话标识
»»» connect_speed	integer	true	无	连接超时	连接超时时间（ms）
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值（ms）
»»» security	string	true	无	验证方式	"none" 或 "token"
»»» token	string	true	无	鉴权Token	security=token 时的鉴权令牌
»»» rtmp_server_port	integer	true	无	RTMP端口	RTMP Server 监听端口号

RTP 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识
»»» name	string	true	无	流名称	流的显示名称
»»» type	string	true	无	协议类型	流协议类型（rtp）
»»» status	integer	true	无	状态	拉流状态
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式
»»» sdp_content	string	true	无	SDP内容	RTP 会话的 SDP 描述内容
»»» connect_speed	integer	true	无	连接超时	连接超时时间（ms）
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值（ms）

UDP 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识
»»» name	string	true	无	流名称	流的显示名称
»»» type	string	true	无	协议类型	流协议类型（udp）
»»» url	string	true	无	流地址	UDP 拉流地址
»»» status	integer	true	无	状态	拉流状态
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式
»»» connect_speed	integer	true	无	连接超时	连接超时时间（ms）
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值（ms）

Zixi 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识
»»» name	string	true	无	流名称	流的显示名称

名称	类型	必选	约束	中文名	说明
»»» type	string	true	无	协议类型	流协议类型（zixi）
»»» url	string	true	无	流地址	Zixi 拉流地址
»»» status	integer	true	无	状态	拉流状态
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式
»»» connect_speed	integer	true	无	连接超时	连接超时时间（ms）
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值（ms）
»»» user	string	true	无	鉴权账号	Zixi 鉴权用户名
»»» password	string	true	无	鉴权密码	Zixi 鉴权密码

HTTP / HTTPS 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识
»»» name	string	true	无	流名称	流的显示名称
»»» type	string	true	无	协议类型	流协议类型（http/https）
»»» url	string	true	无	流地址	HTTP/HTTPS 拉流地址
»»» status	integer	true	无	状态	拉流状态
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式
»»» connect_speed	integer	true	无	连接超时	连接超时时间（ms）
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值（ms）

NDI 协议返回字段

名称	类型	必选	约束	中文名	说明
»»» id	string	true	无	流ID	流的唯一标识

名称	类型	必选	约束	中文名	说明
»»» name	string	true	无	流名称	NDI 源名称
»»» type	string	true	无	协议类型	流协议类型 (ndi)
»»» status	integer	true	无	状态	拉流状态
»»» buffer	string	true	无	缓存策略	缓存策略配置字符串
»»» enable	integer	true	无	启用标志	保留字段
»»» preview_mode	string	true	无	加载模式	解码模式
»»» ndi_name	string	true	无	NDI全名	NDI 源的完整名称
»»» trans_mode	string	true	无	NDI传输模式	"multicast"、"tcp"、"unicast" 或 "rudp"
»»» connect_speed	integer	true	无	连接超时	连接超时时间 (ms)
»»» audio_sync_compst	integer	true	无	音频同步补偿	音频同步补偿值 (ms)
»»» dis_g_name	string	true	无	NDI组名	NDI 发现组名
»»» dis_g_ip	string	true	无	NDI发现IP	NDI 发现服务器 IP

通用恒定字段（所有流类型均返回）

名称	类型	必选	约束	中文名	说明
»»» audio_track_mode	integer	true	无	音轨模式	音频轨道选取模式
»»» audio_track_indices	[integer]	true	无	音轨索引	选中的音频轨道索引数组
»»» audio_track_pids	[integer]	true	无	音轨PID	选中的音频轨道 PID 数组
»»» ts_program_num	integer	true	无	TS节目号	TS 流节目号，无则为 -1
» result	string	true	无	返回结果	请求结果，"ok" 表示成功

返回示例

```
{
  "data": [
    {
      "id": "2ed1f7150cf235e8662d73ac58b925b1",
      "name": "NDI",
      "streams": [
        {
          "address": "192.168.36.238",
          "application": "",

```

```
"audio_sync_compst": 0,
"bandwidth": -1,
"buffer": "live:0:0:0:0:0",
"connect_speed": 0,
"connection_mode": "",
"enable": 0,
"encryption": "",
"id": "0f79952dbc00505b2331b0bdb0c8ca0a",
"latency": -1,
"listener_port": -1,
"name": "N60_238 (HX)",
"ndi_name": "N60_238 (HX)",
"passphrase": "",
"password": "",
"payload_size": -1,
"preview_mode": "Auto",
"rtmp_server_port": -1,
"sdp_content": "",
"security": "",
"session": "public",
"show_advanced": "",
"srt_id": "",
"status": 2,
"token": "",
"trans_mode": "tcp",
"type": "ndi",
"url": "192.168.36.238:5964",
"user": ""
},
{
  "address": "192.168.36.238",
  "application": "",
  "audio_sync_compst": 0,
  "bandwidth": -1,
  "buffer": "live:0:0:0:0:0",
  "connect_speed": 0,
  "connection_mode": "",
  "enable": 0,
  "encryption": "",
  "id": "e893d03a94326b981fd3aa276f90183a",
  "latency": -1,
  "listener_port": -1,
  "name": "N60_238 (HB)",
  "ndi_name": "N60_238 (HB)",
  "passphrase": "",
  "password": "",
  "payload_size": -1,
  "preview_mode": "Auto",
  "rtmp_server_port": -1,
  "sdp_content": "",
  "security": "",
  "session": "public",
  "show_advanced": "",
```

```

        "srt_id": "",
        "status": 2,
        "token": "",
        "trans_mode": "tcp",
        "type": "ndi",
        "url": "192.168.36.238:5965",
        "user": ""
    }
]
},
{
    "id": "0093969758ac42c99245e89f9108ad08",
    "name": "Other",
    "streams": [
        {
            "address": "",
            "application": "",
            "audio_sync_compst": 0,
            "bandwidth": 0,
            "buffer": "live:200:200:500:0:0",
            "connect_speed": 0,
            "connection_mode": "",
            "enable": 0,
            "encryption": "",
            "id": "03db9d3284fd709f6bdd099c9a5f8778",
            "latency": 0,
            "listener_port": 0,
            "name": "RTMP",
            "ndi_name": "",
            "passphrase": "",
            "password": "",
            "payload_size": 0,
            "preview_mode": "Auto",
            "rtmp_server_port": 0,
            "sdp_content": "",
            "security": "",
            "session": "",
            "show_advanced": "",
            "srt_id": "",
            "status": 0,
            "token": "",
            "trans_mode": "",
            "type": "rtmp",
            "url": "rtmp://192.168.0.73:1935/live/vvv",
            "user": ""
        },
        {
            "address": "192.168.36.236",
            "application": "",
            "audio_sync_compst": 0,
            "bandwidth": 25,
            "buffer": "live:200:200:500:0:0",
            "connect_speed": 0,

```

```
"connection_mode": "caller",
"enable": 0,
"encryption": "none",
"id": "834fd00666b91015b23ee336bc2d605a",
"latency": 125,
"listener_port": 1025,
"name": "SRT_Caller",
"ndi_name": "",
"passphrase": "",
"password": "",
"payload_size": 1316,
"preview_mode": "Auto",
"rtmp_server_port": 0,
"sdp_content": "",
"security": "",
"session": "",
"show_advanced": "0",
"srt_id": "",
"status": 0,
"token": "",
"trans_mode": "",
"type": "srt",
"url": "srt://192.168.36.236:1025",
"user": ""
},
{
  "address": "",
  "application": "",
  "audio_sync_compst": 0,
  "bandwidth": 0,
  "buffer": "live:200:200:500:0:0",
  "connect_speed": 0,
  "connection_mode": "",
  "enable": 0,
  "encryption": "",
  "id": "f63dda56f9891ed3b4a5677cf040a4f8",
  "latency": 0,
  "listener_port": 0,
  "name": "RTMP",
  "ndi_name": "",
  "passphrase": "",
  "password": "",
  "payload_size": 0,
  "preview_mode": "Auto",
  "rtmp_server_port": 0,
  "sdp_content": "",
  "security": "",
  "session": "",
  "show_advanced": "",
  "srt_id": "",
  "status": 0,
  "token": "",
  "trans_mode": "",
```

```
        "type": "rtmp",
        "url": "rtmp://192.168.0.73:1935/live/vvv",
        "user": ""
    }
]
},
"result": "ok"
}
```

5.2 POST /source/groups/add

功能说明

创建一个新的视频源分组。需提供组名称，组 ID 由系统自动生成。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» name	body	string	是	分组名称	新分组的显示名称

Body 请求参数

```
{
  "name": "group1"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

5.3 POST /source/groups/streams/add

功能说明

向指定分组中添加一条视频流。根据协议类型（type）的不同，需要携带不同的参数字段。服务端强校验 group_id 和 type，name 在 type≠ndi 时也为必填。

公共请求参数

所有协议类型均需携带以下公共参数：

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌
body	body	object	否	请求体	请求体对象
» group_id	body	string	是	分组ID	目标分组唯一标识
» type	body	string	是	协议类型	流类型：rtmp、rtmps、rtsp、udp、http、https、srt、rtp、ndi、zixi、rtmp_server
» name	body	string	type≠ndi 时必填	流名称	流的显示名称（ndi 类型时可为空，服务端会自动填充）

注意：服务端强校验字段为 group_id 和 type，name 在 type≠ndi 时必填。

RTSP 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	是	流地址	RTSP 拉流地址，格式： rtsp://ip:port/path
» buffer	body	string	否	缓存策略	缓存策略配置字符串，格式： live:min:max:...
» trans_mode	body	string	否	传输模式	RTSP 底层传输协议："tcp" 或 "udp"
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» user	body	string	否	鉴权账号	RTSP 鉴权用户名，无鉴权时留空
» password	body	string	否	鉴权密码	RTSP 鉴权密码，无鉴权时留空

RTMP 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	是	流地址	RTMP 拉流地址，格式： <code>rtmp://ip:port/app/stream</code>
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» user	body	string	否	鉴权账号	RTMP 鉴权用户名，无鉴权时留空
» password	body	string	否	鉴权密码	RTMP 鉴权密码，无鉴权时留空

RTMPS 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	是	流地址	RTMPS 拉流地址，格式： <code>rtmps://ip:port/app/stream</code>
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» user	body	string	否	鉴权账号	RTMPS 鉴权用户名，无鉴权时留空
» password	body	string	否	鉴权密码	RTMPS 鉴权密码，无鉴权时留空

SRT 协议参数

SRT 支持三种连接模式，通过 `connection_mode` 区分：

名称	位置	类型	必选	中文名	说明
» url	body	string	是	流地址	SRT 拉流地址，Listener 模式格式： <code>srt://:port</code> ，Caller/Rendezvous 模式格式： <code>srt://ip:port</code>
» connection_mode	body	string	是	连接模式	Listener（监听模式）、Caller（主叫模式）或 Rendezvous（rendezvous 模式）
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» latency	body	integer	否	延迟	SRT 数据包传输延迟（ms）
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» address	body	string	否	对端IP	Caller/Rendezvous 模式下的对端 IP 地址
» listener_port	body	integer	否	监听端口	本端监听端口号
» srt_id	body	string	否	SRT流ID	SRT 协议自定义流标识
» encryption	body	string	否	加密方式	加密算法："none"、"AES-128"、"AES-192" 或 "AES-256"
» passphrase	body	string	否	密钥	加密密钥字符串，encryption≠none 时必填
» bandwidth	body	integer	否	带宽开销	恢复带宽开销百分比
» payload_size	body	integer	否	载荷大小	SRT 载荷大小（字节）
» show_advanced	body	string	否	显示高级	是否显示高级设置："0" 或 "1"

RTMP Server 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	是	流地址	RTMP Server 推流地址，格式： <code>rtmp://ip:port/app/xxx/stream</code>
» buffer	body	string	否	缓存策略	缓存策略配置字符串

名称	位置	类型	必选	中文名	说明
» application	body	string	否	应用名称	RTMP 应用名称
» session	body	string	否	会话名称	RTMP 会话标识
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» security	body	string	否	验证方式	安全验证方式："none" 或 "token"
» token	body	string	否	鉴权 Token	security=token 时的鉴权令牌
» enable	body	integer	否	启用标志	保留字段
» rtmp_server_port	body	integer	否	RTMP端口	RTMP Server 监听端口号

RTP 协议参数

名称	位置	类型	必选	中文名	说明
» sdp_content	body	string	是	SDP内容	RTP 会话的 SDP 描述内容
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

UDP 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	是	流地址	UDP 拉流地址，格式： <code>udp://ip:port</code> 或直接使用 RTSP 地址
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

Zixi 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	是	流地址	Zixi 拉流地址，格式： zixi://ip:port/stream_id
» stream_id	body	string	否	流ID	Zixi 自定义流标识，若重复则返回 204
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» zixi_udp_out_port	body	integer	否	UDP输出端口	Zixi UDP 输出端口号
» zixi_max_latency	body	integer	否	最大延迟	Zixi 最大延迟（ms）
» encryption	body	string	否	加密方式	加密算法："none"、"AES-128"、"AES-192"或"AES-256"
» passphrase	body	string	否	密钥	加密密钥字符串
» zixi_stream_provider_id	body	string	否	流提供者ID	Zixi 流提供者标识
» zixi_back_url	body	string	否	备用URL	Zixi 备用拉流地址
» zixi_fec_overhead	body	integer	否	FEC开销	前向纠错开销百分比
» zixi_fec_block_ms	body	integer	否	FEC块大小	前向纠错块大小（ms）
» user	body	string	否	鉴权账号	Zixi 鉴权用户名
» password	body	string	否	鉴权密码	Zixi 鉴权密码

HTTP 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	是	流地址	HTTP 拉流地址，格式： http://ip:port/path
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

HTTPS 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	是	流地址	HTTPS 拉流地址，格式： <code>https://ip:port/path</code>
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

NDI 协议参数

NDI 添加时使用 data 数组传递多个 NDI 源，不使用 url/name 字段。data 数组中每项参数如下：

名称	位置	类型	必选	中文名	说明
»» data	body	[object]	是	NDI源列表	NDI 源信息数组，每项为一个 NDI 源
»»» channel	body	string	是	声道	NDI 声道标识，如 "HB"
»»» group_name	body	string	是	发现分组	NDI 发现分组名称，如 "public"
»»» ip	body	string	是	IP地址	NDI 源 IP 地址
»»» name	body	string	是	源名称	NDI 源名称
»»» ndi_name	body	string	是	NDI全名	NDI 源的完整名称
»»» port	body	integer	是	端口号	NDI 源端口号
»»» url	body	string	是	连接地址	NDI 源连接地址，格式： <code>ip:port</code>
»»» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
»»» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

Body 请求参数

RTSP 添加：

```
{
  "group_id": "05b8ba7a42f01af957d2e8483857962f",
  "name": "rtsp1",
  "type": "rtsp",
  "url": "rtsp://192.168.36.222:554/ch01",
  "buffer": "live:0:0:0:0:0",
  "trans_mode": "tcp",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "user": "",
  "password": ""
}
```

RTMP 添加:

```
{
  "group_id": "d054f1d80ed71583392fada325a02823",
  "name": "rtmp1",
  "type": "rtmp",
  "url": "rtmp://192.168.0.73:1935/live/vv",
  "buffer": "live:0:0:0:0:0",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "user": "",
  "password": ""
}
```

RTMPS 添加:

```
{
  "group_id": "d054f1d80ed71583392fada325a02823",
  "name": "rtmps",
  "type": "rtmps",
  "url": "rtmps://192.168.0.73:1935/live/vv",
  "buffer": "live:0:0:0:0:0",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "user": "",
  "password": ""
}
```

SRT Listener 添加:

```
{
  "group_id": "8eca38e84020aba241ead45094c7dd24",
  "name": "srt_listener",
  "type": "srt",
  "url": "srt://:1025",
  "buffer": "live:0:0:0:0:0",
  "connection_mode": "Listener",
  "listener_port": 1025,
}
```

```
"latency": 125,  
"connect_speed": 5000,  
"audio_sync_compst": 0,  
"show_advanced": "1",  
"srt_id": "111",  
"encryption": "none",  
"passphrase": "",  
"bandwidth": 25,  
"payload_size": 1316  
}
```

SRT Rendezvous 添加:

```
{  
  "group_id": "52005872cd91b13767660d196c463879",  
  "name": "srt3",  
  "type": "srt",  
  "url": "srt://192.168.36.51:1025",  
  "buffer": "live:0:0:0:0:0",  
  "connection_mode": "Rendezvous",  
  "address": "192.168.36.51",  
  "listener_port": 1025,  
  "latency": 125,  
  "connect_speed": 5000,  
  "audio_sync_compst": 0,  
  "show_advanced": "1",  
  "srt_id": "111",  
  "encryption": "none",  
  "passphrase": "",  
  "bandwidth": 25,  
  "payload_size": 1316  
}
```

SRT Caller 添加:

```
{  
  "group_id": "8eca38e84020aba241ead45094c7dd24",  
  "name": "srt_caller",  
  "type": "srt",  
  "url": "srt://192.168.36.59:1025",  
  "buffer": "live:0:0:0:0:0",  
  "connection_mode": "Caller",  
  "address": "192.168.36.51",  
  "listener_port": 1025,  
  "latency": 125,  
  "connect_speed": 5000,  
  "audio_sync_compst": 0,  
  "show_advanced": "1",  
  "srt_id": "111",  
  "encryption": "none",  
  "passphrase": "",  
  "bandwidth": 25,  
}
```

```
"payload_size": 1316
}
```

RTMP Server 添加:

```
{
  "group_id": "8eca38e84020aba241ead45094c7dd24",
  "name": "rtmp_server1",
  "type": "rtmp_server",
  "url": "rtmp://192.168.36.59:1935/live/xxx/11111",
  "buffer": "live:0:0:0:0",
  "application": "live",
  "session": "11111",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "security": "token",
  "token": "xxx",
  "enable": 1,
  "rtmp_server_port": 1935
}
```

RTP 添加:

```
{
  "group_id": "e64fbc915a0c34c85cb83bac1e1650fe",
  "name": "rtp_no_audio",
  "type": "rtp",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "buffer": "live:200:200:500:0:0",
  "sdp_content": "SDP:\nv=0\no=- 0 0 IN IP4 127.0.0.1\ns=No Name\nc=IN IP4
192.168.36.197\nt=0 0\na=tool:libavformat 61.1.100\nm=video 1234 RTP/AVP 96\na=rtpmap:96
H264/90000\na=fmtp:96 packetization-mode=1"
}
```

UDP 添加:

```
{
  "group_id": "05b8ba7a42f01af957d2e8483857962f",
  "name": "udp",
  "type": "udp",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "url": "rtsp://192.168.36.222:554/ch01",
  "buffer": "live:0:0:0:0:0"
}
```

Zixi 添加:

```
{
  "group_id": "09c474c50c88be6458464870105a2ed7",
```

```
"name": "zixi",
"type": "zixi",
"url": "zixi://demo.zixi.com:2077/clocks",
"stream_id": "",
"buffer": "live:200:200:500:0:0",
"connect_speed": 5000,
"audio_sync_compst": 0,
"zixi_udp_out_port": 2077,
"zixi_max_latency": 500,
"encryption": "none",
"passphrase": "",
"zixi_stream_provider_id": "clocks",
"zixi_back_url": "",
"zixi_fec_overhead": 30,
"zixi_fec_block_ms": 50,
"user": "",
"password": "",
"newUrl": "zixi://demo.zixi.com:2077/clocks"
}
```

HTTP 添加:

```
{
  "group_id": "05b8ba7a42f01af957d2e8483857962f",
  "name": "http",
  "type": "http",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "url": "http://192.168.36.222:554/ch01",
  "buffer": "live:0:0:0:0:0"
}
```

HTTPS 添加:

```
{
  "group_id": "05b8ba7a42f01af957d2e8483857962f",
  "name": "https",
  "type": "https",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "url": "https://192.168.36.222:554/ch01",
  "buffer": "live:0:0:0:0:0"
}
```

NDI 添加:

```
{
  "group_id": "c719a3c5803d9434050f6ede09d29215",
  "type": "ndi",
  "data": [
    {
```

```
    "channel": "HB",
    "group_name": "public",
    "ip": "192.168.36.238",
    "name": "N60_238 ",
    "ndi_name": "N60_238 (HB)",
    "connect_speed": 5000,
    "audio_sync_compst": 0,
    "port": 5965,
    "url": "192.168.36.238:5965"
  }
]
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

5.4 POST /source/groups/rename

功能说明

重命名指定的视频源分组。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» group_id	body	string	是	分组ID	目标分组的唯一标识
» name	body	string	是	新名称	分组的新显示名称

Body 请求参数

```
{
  "group_id": "1",
  "name": "new_group1"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

5.5 POST /source/groups/remove

功能说明

删除指定的视频源分组及其组内全部流。删除操作不可逆。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» group_id	body	string	是	分组ID	待删除分组的唯一标识

Body 请求参数

```
{
  "group_id": "string"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

5.6 POST /source/groups/reorder

功能说明

调整视频源分组的显示顺序。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» data	body	array	是	分组顺序	分组ID数组，数组顺序即新的显示顺序，须包含全部现有分组
»» group_id	body	string	是	分组ID	分组唯一标识

Body 请求参数

```
{
  "data": [
    {"group_id": "2"},
    {"group_id": "1"},
    {"group_id": "3"}
  ]
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

5.7 POST /source/groups/streams/modify

功能说明

修改指定视频流的参数。服务端通过 `stream_id` 全局定位流，`type` 为强校验字段，其余字段按需提交即可。

公共请求参数

所有协议类型均需携带以下公共参数：

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌
body	body	object	否	请求体	请求体对象
» stream_id	body	string	是	流ID	待修改流的唯一标识
» type	body	string	是	协议类型	流类型：rtmp、rtmps、rtsp、udp、http、https、srt、rtp、ndi、zixi、rtmp_server
» name	body	string	否	流名称	流的新显示名称

注意：服务端强校验字段为 `stream_id` 和 `type`；其余字段按需提交即可。

RTSP 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	否	流地址	RTSP 拉流地址
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» trans_mode	body	string	否	传输模式	RTSP 底层传输协议："tcp" 或 "udp"
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» user	body	string	否	鉴权账号	RTSP 鉴权用户名，无鉴权时留空
» password	body	string	否	鉴权密码	RTSP 鉴权密码，无鉴权时留空

RTMP 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	否	流地址	RTMP 拉流地址
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» user	body	string	否	鉴权账号	RTMP 鉴权用户名，无鉴权时留空
» password	body	string	否	鉴权密码	RTMP 鉴权密码，无鉴权时留空

RTMPS 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	否	流地址	RTMPS 拉流地址
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» user	body	string	否	鉴权账号	RTMPS 鉴权用户名，无鉴权时留空
» password	body	string	否	鉴权密码	RTMPS 鉴权密码，无鉴权时留空

SRT 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	否	流地址	SRT 拉流地址
» connection_mode	body	string	否	连接模式	Listener、Caller 或 Rendezvous
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» latency	body	integer	否	延迟	SRT 数据包传输延迟（ms）
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» address	body	string	否	对端IP	Caller/Rendezvous 模式下的对端 IP 地址
» listener_port	body	integer	否	监听端口	本端监听端口号

名称	位置	类型	必选	中文名	说明
» srt_id	body	string	否	SRT流ID	SRT 协议自定义流标识
» encryption	body	string	否	加密方式	"none"、"AES-128"、"AES-192" 或 "AES-256"
» passphrase	body	string	否	密钥	加密密钥字符串
» bandwidth	body	integer	否	带宽开销	恢复带宽开销百分比
» payload_size	body	integer	否	载荷大小	SRT 载荷大小（字节）
» show_advanced	body	string	否	显示高级	是否显示高级设置："0" 或 "1"

RTMP Server 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	否	流地址	RTMP Server 推流地址
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» application	body	string	否	应用名称	RTMP 应用名称
» session	body	string	否	会话名称	RTMP 会话标识
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» security	body	string	否	验证方式	"none" 或 "token"
» token	body	string	否	鉴权Token	security=token 时的鉴权令牌
» enable	body	integer	否	启用标志	保留字段
» rtmp_server_port	body	integer	否	RTMP端口	RTMP Server 监听端口号

RTP 协议参数

名称	位置	类型	必选	中文名	说明
» sdp_content	body	string	否	SDP内容	RTP 会话的 SDP 描述内容
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

UDP 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	否	流地址	UDP 拉流地址
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

Zixi 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	否	流地址	Zixi 拉流地址
» stream_id	body	string	否	流ID	Zixi 自定义流标识
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）
» encryption	body	string	否	加密方式	"none"、"AES-128"、"AES-192" 或 "AES-256"
» passphrase	body	string	否	密钥	加密密钥字符串
» user	body	string	否	鉴权账号	Zixi 鉴权用户名
» password	body	string	否	鉴权密码	Zixi 鉴权密码

HTTP 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	否	流地址	HTTP 拉流地址
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

HTTPS 协议参数

名称	位置	类型	必选	中文名	说明
» url	body	string	否	流地址	HTTPS 拉流地址
» buffer	body	string	否	缓存策略	缓存策略配置字符串
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

NDI 协议参数

名称	位置	类型	必选	中文名	说明
» ndi_name	body	string	否	NDI名称	NDI 源的完整名称
» trans_mode	body	string	否	NDI传输模式	NDI 传输模式："multicast"、"tcp"、"unicast" 或 "rdp"
» buffer	body	string	否	缓存策略	NDI 缓存策略配置（长度超过 1000 将被拒绝，返回 207）
» connect_speed	body	integer	否	连接超时	连接超时时间（ms）
» audio_sync_compst	body	integer	否	音频同步补偿	音频同步补偿值（ms）

Body 请求参数

修改 RTMP：

```
{
  "stream_id": "d054f1d80ed71583392fada325a02823",
  "name": "rtmp1",
  "type": "rtmp",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "url": "rtmp://192.168.0.73:1935/live/vv",
  "buffer": "live:0:0:0:0:0",
  "user": "",
  "password": ""
}
```

修改 RTMPS：

```
{
  "stream_id": "d054f1d80ed71583392fada325a02823",
  "name": "rtmps",
  "type": "rtmps",
  "url": "rtmps://192.168.0.73:1935/live/vv",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "buffer": "live:0:0:0:0:0",
  "user": "",
  "password": ""
}
```

修改 UDP:

```
{
  "stream_id": "05b8ba7a42f01af957d2e8483857962f",
  "name": "udp",
  "type": "udp",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "url": "udp://192.168.36.222:554/ch01",
  "buffer": "live:0:0:0:0:0"
}
```

修改 HTTP:

```
{
  "stream_id": "05b8ba7a42f01af957d2e8483857962f",
  "name": "http",
  "type": "http",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "url": "http://192.168.36.222:554/ch01",
  "buffer": "live:0:0:0:0:0"
}
```

修改 HTTPS:

```
{
  "stream_id": "05b8ba7a42f01af957d2e8483857962f",
  "name": "https",
  "type": "https",
  "url": "https://192.168.36.222:554/ch01",
  "buffer": "live:0:0:0:0:0",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "user": "",
  "password": ""
}
```

修改 RTSP:

```
{
  "stream_id": "05b8ba7a42f01af957d2e8483857962f",
  "name": "rtsp1",
  "type": "rtsp",
  "url": "rtsp://192.168.36.222:554/ch01",
  "buffer": "live:0:0:0:0",
  "trans_mode": "tcp",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "user": "",
  "password": ""
}
```

修改 RTP:

```
{
  "stream_id": "8ef8bd5971763f671d81efa994549764",
  "name": "rtp1",
  "type": "rtp",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "buffer": "live:0:0:0:0",
  "sdp_content": "SDP:\nv=0\no=- 0 0 IN IP4 127.0.0.1\ns=No Name\nc=IN IP4
192.168.36.197\n\t=0 0\na=tool:libavformat 61.1.100\nm=audio 1236 RTP/AVP
97\nb=AS:768\na=rtpmap:97 PCMU/48000/2"
}
```

修改 SRT (Listener) :

```
{
  "stream_id": "8eca38e84020aba241ead45094c7dd24",
  "name": "srt_listener",
  "type": "srt",
  "url": "srt://:1025",
  "buffer": "live:0:0:0:0",
  "connection_mode": "Listener",
  "listener_port": 1025,
  "latency": 125,
  "show_advanced": "1",
  "srt_id": "111",
  "encryption": "none",
  "passphrase": "",
  "bandwidth": 25,
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "payload_size": 1316
}
```

修改 SRT (Caller) :

```
{
```

```

"stream_id": "8eca38e84020aba241ead45094c7dd24",
"name": "srt_caller",
"type": "srt",
"url": "srt://192.168.36.59:1025",
"buffer": "live:0:0:0:0:0",
"connection_mode": "Caller",
"address": "192.168.36.51",
"listener_port": 1025,
"latency": 125,
"show_advanced": "1",
"srt_id": "111",
"encryption": "none",
"passphrase": "",
"connect_speed": 5000,
"audio_sync_compst": 0,
"bandwidth": 25,
"payload_size": 1316
}

```

修改 SRT (Rendezvous) :

```

{
  "stream_id": "52005872cd91b13767660d196c463879",
  "name": "srt3",
  "type": "srt",
  "url": "srt://192.168.36.51:1025",
  "buffer": "live:0:0:0:0:0",
  "connection_mode": "Rendezvous",
  "address": "192.168.36.51",
  "listener_port": 1025,
  "latency": 125,
  "show_advanced": "1",
  "srt_id": "111",
  "encryption": "none",
  "passphrase": "",
  "bandwidth": 25,
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "payload_size": 1316
}

```

修改 RTMP Server:

```

{
  "stream_id": "8eca38e84020aba241ead45094c7dd24",
  "name": "rtmp_server1",
  "type": "rtmp_server",
  "url": "rtmp://192.168.36.59:1935/live/xxx/11111",
  "buffer": "live:0:0:0:0:0",
  "application": "live",
  "session": "11111",
  "security": "token",
}

```

```
"token": "xxx",
"connect_speed": 5000,
"audio_sync_compst": 0,
"enable": 1,
"rtmp_server_port": 1935
}
```

修改 NDI:

```
{
  "stream_id": "c0343e3a2fed40071b1b51a4af53f3c3",
  "type": "ndi",
  "url": "192.168.36.155:5963",
  "buffer": "live:1000:1000:1500:0:0",
  "trans_mode": "multicast",
  "ndi_name": "N6-21-1080P60 (Channel-NDI)",
  "connect_speed": 5000,
  "audio_sync_compst": 0,
  "name": "N6-21-1080P60",
  "preview_mode": "Full"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

5.8 POST /source/groups/streams/remove

功能说明

删除指定分组中的某一条视频流。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» group_id	body	string	是	分组ID	流所属分组的唯一标识

名称	位置	类型	必选	中文名	说明
» stream_id	body	string	是	流ID	待删除流的唯一标识

Body 请求参数

```
{
  "group_id": "1",
  "stream_id": "1"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

5.9 POST /source/groups/streams/copy

功能说明

将指定视频流从源分组复制到目标分组。新流将获得新的流 ID，源流保持不变。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» from_group_id	body	string	是	源分组ID	复制来源分组的唯一标识
» from_stream_id	body	string	是	源流ID	待复制流的唯一标识
» to_group_id	body	string	是	目标分组ID	复制目标分组的唯一标识

Body 请求参数

```
{
  "from_group_id": "83b50ca8e8af2ec13f35b251b50076dd",
  "from_stream_id": "17dd78a37db89a20eb4a11a7705dafb2",
  "to_group_id": "354ee7bf5984462d483151fcc98f74fb"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

5.10 POST /source/groups/streams/move

功能说明

将指定视频流从源分组迁移到目标分组。移动后流将属于目标分组，源分组中不再包含该流。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» from_group_id	body	string	是	源分组ID	移动来源分组的唯一标识
» from_stream_id	body	string	是	源流ID	待移动流的唯一标识
» to_group_id	body	string	是	目标分组ID	移动目标分组的唯一标识

Body 请求参数

```
{
  "from_group_id": "83b50ca8e8af2ec13f35b251b50076dd",
  "from_stream_id": "17dd78a37db89a20eb4a11a7705dafb2",
  "to_group_id": "354ee7bf5984462d483151fcc98f74fb"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

6. LAYOUT模块

LAYOUT（布局）模块用于管理多画面布局模板。布局模板定义了输出画面的窗格划分方式（位置、大小、层级），每个窗格可绑定一个视频源流，支持多种预设布局以及自定义布局的创建、修改、保存、选择等操作。

6.1 GET /layout/template/list

功能说明

获取系统内置的布局模板列表。每个模板定义了窗格的数量、位置和尺寸比例，作为创建自定义布局的基础。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» count	integer	true	无	模板总数	模板模板的总数量
» data	[object]	true	无	模板列表	布局模板数组
»» name	string	true	无	布局名称	模板的名称，如 "Single"、"PIP"、"2x2" 等
»» position	[object]	true	无	窗格位置	模板中各窗格的位置和尺寸信息
»»» height	number	true	无	高度	窗格高度，以百分比表示（浮点数，如 33.33）
»»» id	integer	true	无	窗格ID	窗格唯一标识
»»» number	integer	true	无	窗格序号	窗格在模板中的序号位置
»»» stream_id	string	true	无	绑定源ID	当前窗格绑定的信号源 ID（模板中为空字符串）

名称	类型	必选	约束	中文名	说明
»»» width	number	true	无	宽度	窗格宽度，以百分比表示（浮点数）
»»» x	number	true	无	X坐标	窗格左上角 X 轴位置，以百分比表示（浮点数）
»»» y	number	true	无	Y坐标	窗格左上角 Y 轴位置，以百分比表示（浮点数）
»»» z_index	integer	true	无	层级	窗格 Z 轴层级
»» template_count	integer	true	无	模板窗格数	该模板包含的窗格总数
»» template_id	integer	true	无	模板ID	模板唯一标识
» result	string	true	无	操作结果	请求结果，"ok" 表示成功

返回示例

```
{
  "count": 1000,
  "data": [
    {
      "name": "Single",
      "position": [
        {
          "height": 100,
          "id": 1,
          "number": 1,
          "stream_id": "",
          "width": 100,
          "x": 0,
          "y": 0,
          "z_index": 0
        }
      ],
      "template_count": 1,
      "template_id": 1
    },
    {
      "name": "PIP",
      "position": [
        {
          "height": 100,
          "id": 1,
          "number": 1,
          "stream_id": "",
          "width": 100,
          "x": 0,
          "y": 0,
```

```
    "z_index": 0
  },
  {
    "height": 30,
    "id": 2,
    "number": 2,
    "stream_id": "",
    "width": 30,
    "x": 69,
    "y": 69,
    "z_index": 1
  }
],
"template_count": 2,
"template_id": 2
},
{
  "name": "2x2",
  "position": [
    {
      "height": 50,
      "id": 1,
      "number": 1,
      "stream_id": "",
      "width": 50,
      "x": 0,
      "y": 0,
      "z_index": 0
    },
    {
      "height": 50,
      "id": 2,
      "number": 2,
      "stream_id": "",
      "width": 50,
      "x": 50,
      "y": 0,
      "z_index": 1
    },
    {
      "height": 50,
      "id": 3,
      "number": 3,
      "stream_id": "",
      "width": 50,
      "x": 0,
      "y": 50,
      "z_index": 2
    },
    {
      "height": 50,
      "id": 4,
      "number": 4,
```

```
        "stream_id": "",
        "width": 50,
        "x": 50,
        "y": 50,
        "z_index": 3
    }
],
"template_count": 4,
"template_id": 3
},
{
    "name": "3x3",
    "position": [
        {
            "height": 33.33,
            "id": 1,
            "number": 1,
            "stream_id": "",
            "width": 33.33,
            "x": 0,
            "y": 0,
            "z_index": 0
        },
        {
            "height": 33.33,
            "id": 2,
            "number": 2,
            "stream_id": "",
            "width": 33.33,
            "x": 33.33,
            "y": 0,
            "z_index": 1
        },
        {
            "height": 33.33,
            "id": 3,
            "number": 3,
            "stream_id": "",
            "width": 33.33,
            "x": 66.67,
            "y": 0,
            "z_index": 2
        },
        {
            "height": 33.33,
            "id": 4,
            "number": 4,
            "stream_id": "",
            "width": 33.33,
            "x": 0,
            "y": 33.33,
            "z_index": 3
        }
    ],
}
```

```
{
  "height": 33.33,
  "id": 5,
  "number": 5,
  "stream_id": "",
  "width": 33.33,
  "x": 33.33,
  "y": 33.33,
  "z_index": 4
},
{
  "height": 33.33,
  "id": 6,
  "number": 6,
  "stream_id": "",
  "width": 33.33,
  "x": 66.67,
  "y": 33.33,
  "z_index": 5
},
{
  "height": 33.33,
  "id": 7,
  "number": 7,
  "stream_id": "",
  "width": 33.33,
  "x": 0,
  "y": 66.67,
  "z_index": 6
},
{
  "height": 33.33,
  "id": 8,
  "number": 8,
  "stream_id": "",
  "width": 33.33,
  "x": 33.33,
  "y": 66.67,
  "z_index": 7
},
{
  "height": 33.33,
  "id": 9,
  "number": 9,
  "stream_id": "",
  "width": 33.33,
  "x": 66.67,
  "y": 66.67,
  "z_index": 8
}
],
"template_count": 9,
"template_id": 4
```

```
},
{
  "name": "4x4",
  "position": [
    {
      "height": 25,
      "id": 1,
      "number": 1,
      "stream_id": "",
      "width": 25,
      "x": 0,
      "y": 0,
      "z_index": 0
    },
    {
      "height": 25,
      "id": 2,
      "number": 2,
      "stream_id": "",
      "width": 25,
      "x": 25,
      "y": 0,
      "z_index": 1
    },
    {
      "height": 25,
      "id": 3,
      "number": 3,
      "stream_id": "",
      "width": 25,
      "x": 50,
      "y": 0,
      "z_index": 2
    },
    {
      "height": 25,
      "id": 4,
      "number": 4,
      "stream_id": "",
      "width": 25,
      "x": 75,
      "y": 0,
      "z_index": 3
    },
    {
      "height": 25,
      "id": 5,
      "number": 5,
      "stream_id": "",
      "width": 25,
      "x": 0,
      "y": 25,
      "z_index": 4
    }
  ]
}
```

```
},
{
  "height": 25,
  "id": 6,
  "number": 6,
  "stream_id": "",
  "width": 25,
  "x": 25,
  "y": 25,
  "z_index": 5
},
{
  "height": 25,
  "id": 7,
  "number": 7,
  "stream_id": "",
  "width": 25,
  "x": 50,
  "y": 25,
  "z_index": 6
},
{
  "height": 25,
  "id": 8,
  "number": 8,
  "stream_id": "",
  "width": 25,
  "x": 75,
  "y": 25,
  "z_index": 7
},
{
  "height": 25,
  "id": 9,
  "number": 9,
  "stream_id": "",
  "width": 25,
  "x": 0,
  "y": 50,
  "z_index": 8
},
{
  "height": 25,
  "id": 10,
  "number": 10,
  "stream_id": "",
  "width": 25,
  "x": 25,
  "y": 50,
  "z_index": 9
},
{
  "height": 25,
```

```
"id": 11,  
"number": 11,  
"stream_id": "",  
"width": 25,  
"x": 50,  
"y": 50,  
"z_index": 10  
},  
{  
  "height": 25,  
  "id": 12,  
  "number": 12,  
  "stream_id": "",  
  "width": 25,  
  "x": 75,  
  "y": 50,  
  "z_index": 11  
},  
{  
  "height": 25,  
  "id": 13,  
  "number": 13,  
  "stream_id": "",  
  "width": 25,  
  "x": 0,  
  "y": 75,  
  "z_index": 12  
},  
{  
  "height": 25,  
  "id": 14,  
  "number": 14,  
  "stream_id": "",  
  "width": 25,  
  "x": 25,  
  "y": 75,  
  "z_index": 13  
},  
{  
  "height": 25,  
  "id": 15,  
  "number": 15,  
  "stream_id": "",  
  "width": 25,  
  "x": 50,  
  "y": 75,  
  "z_index": 14  
},  
{  
  "height": 25,  
  "id": 16,  
  "number": 16,  
  "stream_id": "",
```

```
        "width": 25,
        "x": 75,
        "y": 75,
        "z_index": 15
      }
    ],
    "template_count": 16,
    "template_id": 5
  }
],
"result": "ok"
}
```

6.2 GET /layout/list

功能说明

获取当前用户已创建的所有自定义布局列表，包括每个布局的窗格配置和信号源绑定状态。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	布局列表	自定义布局数组
»» layout_count	integer	true	无	窗格数量	该布局包含的窗格总数
»» layout_id	integer	true	无	布局ID	布局唯一标识
»» name	string	true	无	布局名称	布局用户自定义名称
»» position	[object]	true	无	窗格位置	布局中各窗格的位置和尺寸信息
»»» height	number	true	无	高度	窗格高度，以百分比表示（浮点数，如33.33）
»»» id	integer	true	无	窗格ID	窗格唯一标识
»»» number	integer	true	无	窗格序号	窗格在布局中的序号位置

名称	类型	必选	约束	中文名	说明
»»» stream_id	string	true	无	绑定源 ID	窗格当前绑定的信号源 ID（未绑定则为空字符串）
»»» width	number	true	无	宽度	窗格宽度，以百分比表示（浮点数）
»»» x	number	true	无	X坐标	窗格左上角 X 轴位置，以百分比表示（浮点数）
»»» y	number	true	无	Y坐标	窗格左上角 Y 轴位置，以百分比表示（浮点数）
»»» z_index	integer	true	无	层级	窗格 Z 轴层级
»» used	boolean	true	无	已启用	该布局是否已被某个输出通道使用
» result	string	true	无	操作结果	请求结果，"ok" 表示成功

返回示例

```
{
  "data": [
    {
      "layout_count": 1,
      "layout_id": 1,
      "name": "Single",
      "position": [
        {
          "height": 100,
          "id": 1,
          "number": 1,
          "stream_id": "",
          "width": 100,
          "x": 0,
          "y": 0,
          "z_index": 0
        }
      ],
      "used": false
    },
    {
      "layout_count": 2,
      "layout_id": 2,
      "name": "PIP",
      "position": [
        {
          "height": 100,
          "id": 1,
          "number": 1,
          "stream_id": "",
```

```
        "width": 100,
        "x": 0,
        "y": 0,
        "z_index": 0
    },
    {
        "height": 30,
        "id": 2,
        "number": 2,
        "stream_id": "",
        "width": 30,
        "x": 69,
        "y": 69,
        "z_index": 1
    }
],
"used": false
},
{
    "layout_count": 4,
    "layout_id": 3,
    "name": "2x2",
    "position": [
        {
            "height": 50,
            "id": 1,
            "number": 1,
            "stream_id": "",
            "width": 50,
            "x": 0,
            "y": 0,
            "z_index": 0
        },
        {
            "height": 50,
            "id": 2,
            "number": 2,
            "stream_id": "",
            "width": 50,
            "x": 50,
            "y": 0,
            "z_index": 1
        },
        {
            "height": 50,
            "id": 3,
            "number": 3,
            "stream_id": "",
            "width": 50,
            "x": 0,
            "y": 50,
            "z_index": 2
        },
        {
            "height": 50,
            "id": 4,
            "number": 4,
            "stream_id": "",
            "width": 50,
            "x": 50,
            "y": 50,
            "z_index": 3
        }
    ]
}
```

```
{
  {
    "height": 50,
    "id": 4,
    "number": 4,
    "stream_id": "",
    "width": 50,
    "x": 50,
    "y": 50,
    "z_index": 3
  }
],
"used": false
}
],
"result": "ok"
}
```

6.3 POST /layout/add

功能说明

基于系统模板创建一个新的自定义布局。需指定模板 ID 和自定义布局名称。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» template_id	body	integer	是	模板ID	基于的模板唯一标识
» name	body	string	是	布局名称	新建布局的自定义名称

Body 请求参数

```
{
  "template_id": 1,
  "name": "AddSingle2"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

6.4 POST /layout/remove

功能说明

删除指定的自定义布局预设。被输出通道使用的布局可能无法删除。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» layout_id	body	integer	是	布局ID	待删除布局的唯一标识

Body 请求参数

```
{
  "layout_id": 7
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

6.5 POST /layout/select

功能说明

将指定布局应用到某个输出通道（output）。选择后该输出将按照布局的窗格配置显示画面。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识

名称	位置	类型	必选	中文名	说明
body	body	object	否	请求体	请求体对象
» output_id	body	string	是	输出ID	目标输出通道标识
» layout_id	body	integer	是	布局ID	待应用的布局唯一标识

Body 请求参数

```
{
  "output_id": "1",
  "layout_id": 2
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

6.6 POST /layout/modify

功能说明

修改指定布局中某个窗格的位置和尺寸参数。修改立即生效于绑定了该布局的输出通道。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» layout_id	body	integer	是	布局ID	目标布局的唯一标识
» position	body	[object]	是	窗格配置	要修改的窗格配置数组
»» width	body	integer	否	宽度	窗格新宽度, 以百分比表示
»» height	body	integer	否	高度	窗格新高度, 以百分比表示
»» number	body	integer	否	窗格序号	窗格的序号位置

名称	位置	类型	必选	中文名	说明
»» z_index	body	integer	否	层级	窗格的新 Z 轴层级
»» id	body	integer	否	窗格ID	目标窗格的唯一标识
»» x	body	integer	否	X坐标	窗格新左上角 X 坐标，以百分比表示
»» y	body	integer	否	Y坐标	窗格新左上角 Y 坐标，以百分比表示
» output_id	body	string	是	输出ID	关联的输出通道标识

Body 请求参数

```
{
  "layout_id": 6,
  "position": [
    {
      "width": 30,
      "height": 30,
      "number": 2,
      "z_index": 1,
      "id": 2,
      "x": 0,
      "y": 0
    }
  ],
  "output_id": "1"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

6.7 POST /layout/save

功能说明

将输出通道中当前的布局窗格修改（包括位置、尺寸变更及信号源绑定）保存到指定布局预设中，覆盖原布局配置。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» output_id	body	string	是	输出ID	输出通道标识，支持 "1" 和 "2"（共两个输出通道）
» layout_id	body	integer	是	布局ID	保存目标布局的唯一标识

Body 请求参数

```
{
  "output_id": "1",
  "layout_id": 6
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果，"ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

6.8 POST /layout/resave

功能说明

将当前输出通道的布局配置另存为一个全新的布局预设。新布局将获得独立的 layout_id，原布局不受影响。

说明：另存为会生成一个全新的布局 ID，服务端不使用传入的 layout_id（即使传了也会被忽略）。只需 output_id 和 layout_name。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» output_id	body	string	是	输出ID	当前输出通道标识

名称	位置	类型	必选	中文名	说明
» layout_name	body	string	是	布局名称	另存后的新布局名称

Body 请求参数

```
{
  "output_id": "1",
  "layout_name": "Test"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

6.9 POST /layout/reload

功能说明

从指定布局预设重新加载输出通道的布局配置，丢弃当前输出通道上所有未保存的布局修改。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» output_id	body	integer	是	输出ID	目标输出通道标识
» layout_id	body	integer	是	布局ID	待重新加载的布局唯一标识
» layout_name	body	string	否	布局名称	布局名称（可选，后端逻辑可能用到）

Body 请求参数

```
{
  "output_id": 1,
  "layout_name": "5",
  "layout_id": 6
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

6.10 POST /layout/rename

功能说明

修改指定布局预设的名称。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	应用标识	请求来源应用标识
body	body	object	否	请求体	请求体对象
» layout_id	body	integer	是	布局ID	目标布局的唯一标识
» name	body	string	是	新名称	布局的新显示名称

Body 请求参数

```
{
  "layout_id": 1,
  "name": ""
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果, "ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

7. OUTPUT模块

OUTPUT（视频输出）模块用于管理设备的视频和音频输出配置。涵盖输出接口参数设置、分辨率与帧率调整、音频混音控制、边框与背景配置、编码参数管理、TS 流节目配置、PTZ 控制以及输出静音和音量指示等功能。

7.1 GET /output/interfaces/get

功能说明

获取输出接口类型列表

请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	否	输出ID	布局ID
type	query	string	否	接口类型	输出类型：video，audio
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

type=video 时 data 数组每项字段：

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	返回数据	视频输出接口列表
»» colorspace	string	true	无	色彩模式	当前输出接口色彩模式
»» enable	boolean	true	无	启用状态	是否开启
»» id	integer	true	无	输出ID	输出接口ID
»» label	string	true	无	标签	输出接口显示名称
»» mode	string	true	无	接口类型	输出接口类型
»» support	boolean	true	无	是否支持	当前设备是否支持当前接口
»» visible	boolean	true	无	可见	输出接口是否可见
» result	string	true	无	返回结果	请求结果状态

type=audio 时 data 数组每项字段 (无 colorspace/mode/support) :

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	返回数据	音频输出接口列表
»» id	integer	true	无	音频接口ID	音频接口ID
»» label	string	true	无	标签	音频接口显示名称
»» enable	boolean	true	无	启用状态	是否开启
»» samplerate	number	true	无	采样率	采样率 (kHz, 如 44.1/48)
»» channels	integer	true	无	通道数	声道数
»» volume	integer	true	无	音量	音量
»» mute	boolean	true	无	静音	是否静音
»» visible	boolean	true	无	可见	接口是否可见
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": [
    {
      "colorspace": "RGB444",
      "enable": false,
      "id": 1,
      "label": "HDMI1",
      "mode": "HDMI",
      "support": false,
      "visible": true
    },
    {
      "colorspace": "RGB444",
      "enable": false,
      "id": 2,
      "label": "HDMI2",
      "mode": "HDMI",
      "support": true,
      "visible": true
    },
    {
      "colorspace": "RGB444",
      "enable": false,
      "id": 3,
      "label": "SDI",
      "mode": "HDMI",
      "support": true,
      "visible": true
    }
  ]
}
```

```
}
],
"result": "ok"
}
{
  "data": [
    {
      "id": 1,
      "label": "HDMI",
      "enable": false,
      "samplerate": 44.1,
      "channels": 2,
      "volume": 0,
      "mute": false,
      "visible": true
    },
    {
      "id": 2,
      "label": "HDMI1",
      "enable": false,
      "samplerate": 44.1,
      "channels": 2,
      "volume": 0,
      "mute": false,
      "visible": true
    },
    {
      "id": 3,
      "label": "SDI",
      "enable": false,
      "samplerate": 44.1,
      "channels": 2,
      "volume": 0,
      "mute": false,
      "visible": true
    },
    {
      "id": 4,
      "label": "LineOut",
      "enable": false,
      "samplerate": 44.1,
      "channels": 2,
      "volume": 0,
      "mute": false,
      "visible": true
    }
  ],
  "result": "ok"
}
```

7.2 POST /output/interfaces/set

功能说明

设置输出接口参数

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» output_id	body	string	是	输出ID	输出ID
» type	body	string	是	接口类型	类型 video 或 audio
» intf_id	body	integer	是	接口ID	接口ID（接口以 intf_id 为准）
» enable	body	boolean	是	启用状态	是否开启
» channels	body	integer	type=audio 时必填	通道数	通道数；intf_id 1/2/4 要求为 2
» samplerate	body	integer	type=audio 时必填	采样率	采样率；intf_id 3 要求为 48000
» volume	body	integer	否	音量	音量（type=audio 生效）
» mute	body	boolean	否	静音	是否静音（type=audio 生效）
» id	body	integer	否	接口ID	忽略（接口以 intf_id 定位）
» label	body	string	否	标签	忽略
» visible	body	boolean	否	可见	忽略

说明：音频接口约束——`intf_id` 为 1/2/4 时 `channels` 必须为 2，`intf_id` 为 3 时 `samplerate` 必须为 48000，否则返回 `{"result":"error","msg":"1"}`（参数错误）。

Body 请求参数

```
{
  "channels": 2,
  "enable": true,
  "id": 1,
  "label": "HDMI1",
  "mute": true,
  "samplerate": 48000,
  "visible": true,
  "volume": 14,
  "output_id": "1",
  "type": "audio",
  "intf_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

7.3 GET /output/list

功能说明

获取窗口列表

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	返回数据	窗口列表数据
»» background	string	true	无	背景颜色	背景颜色
»» bg_enable	boolean	true	无	背景启用	是否开启背景

名称	类型	必选	约束	中文名	说明
»» border	object	true	无	边框参数	边框参数
»»» bottom_width	integer	true	无	下边框宽度	边框宽度
»»» color	string	true	无	边框颜色	边框颜色
»»» enable	boolean	true	无	边框启用	是否开启边框
»»» hex_color	integer	true	无	十六进制颜色值	边框颜色的十六进制数值
»»» left_width	integer	true	无	左边框宽度	边框宽度
»»» right_width	integer	true	无	右边框宽度	边框宽度
»»» top_width	integer	true	无	上边框宽度	边框宽度
»» id	string	true	无	窗格ID	布局ID
»» layout_id	integer	true	无	布局ID	布局ID
»» modified	boolean	true	无	是否已修改	当前布局是否已被修改
»» name	string	true	无	名称	布局名称
»» res_id	string	true	无	分辨率ID	输出分辨率ID
»» res_name	string	true	无	分辨率名称	输出分辨率名称
»» vumeter	boolean	true	无	音量指示器	是否显示音量指示器
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": [
    {
      "background": "#000000",
      "bg_enable": false,
      "border": {
        "bottom_width": 2,
        "color": "#FF0000",
        "enable": false,
        "hex_color": 16711680,
        "left_width": 2,
        "right_width": 2,
        "top_width": 2
      },
      "id": "1",
      "layout_id": 2,
      "modified": false,
      "name": "output1",
      "res_id": "5",
```

```
    "res_name": "1920x1080P60",
    "vumeter": true
  },
  {
    "background": "#000000",
    "bg_enable": false,
    "border": {
      "bottom_width": 2,
      "color": "#FF0000",
      "enable": false,
      "hex_color": 16711680,
      "left_width": 2,
      "right_width": 2,
      "top_width": 2
    },
    "id": "2",
    "layout_id": 1,
    "modified": false,
    "name": "output2",
    "res_id": "5",
    "res_name": "1920x1080P60",
    "vumeter": true
  }
],
"result": "ok"
}
```

7.4 GET /output/get

功能说明

获取某一窗口信息

请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	是	输出ID	布局ID
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	无	返回数据	窗口数据
»» layout_id	integer	true	无	布局ID	布局ID
»» name	string	true	无	名称	输出名称
»» position	[object]	true	无	窗格位置	窗格位置列表

名称	类型	必选	约束	中文名	说明
»»» enable	boolean	true	无	启用状态	是否开启拉流
»»» height	integer	true	无	高度	高度 单位:百分比
»»» id	integer	true	无	窗格ID	位置ID
»»» mute	boolean	true	无	静音	是否静音
»»» number	integer	true	无	序号	位置所处下标
»»» resolution	string	true	无	分辨率	输出分辨率
»»» status	integer	true	无	流状态	流状态，参考流获取定义
»»» stream_id	string	true	无	流ID	当前位置正在加载的流ID
»»» stream_ip	string	true	无	流IP	当前位置正在加载的流IP
»»» stream_name	string	true	无	流名称	当前位置正在加载的流名称
»»» type	string	true	无	流类型	当前位置正在加载的流类型
»»» video_bitrate	string	true	无	码流	当前位置正在加载的流码流
»»» video_format	string	true	无	编码类型	当前位置正在加载的流编码类型
»»» volume	integer	true	无	音量	音量
»»» vu_meter	boolean	true	无	音量指示器	是否显示音量指示器
»»» width	integer	true	无	宽度	宽度 单位:百分比
»»» x	integer	true	无	X轴位置	x轴位置 单位： 百分比
»»» y	integer	true	无	Y轴位置	y轴位置 单位： 百分比
»»» z_index	integer	true	无	Z轴层级	层级顺序
»» vumeter	boolean	true	无	音量指示器	是否显示音量指示器
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": {
    "layout_id": 2,
    "name": "output1",
    "position": [
      {
        "enable": true,
        "height": 30,
        "id": 1,
        "mute": false,
```

```

        "number": 1,
        "resolution": "1920x1080p60",
        "status": 2,
        "stream_id": "e893d03a94326b981fd3aa276f90183a",
        "stream_ip": "192.168.36.238:5965",
        "stream_name": "N60_238 (HB)",
        "type": "ndi",
        "video_bitrate": "55479.4",
        "video_format": "NDI | HB",
        "volume": 0,
        "vu_meter": true,
        "width": 30,
        "x": 0,
        "y": 0,
        "z_index": 0
    },
    {
        "enable": false,
        "height": 30,
        "id": 2,
        "mute": false,
        "number": 2,
        "resolution": "0x0p0",
        "status": 2,
        "stream_id": "",
        "stream_ip": "",
        "stream_name": "",
        "type": "",
        "video_bitrate": "0",
        "video_format": "",
        "volume": 0,
        "vu_meter": true,
        "width": 30,
        "x": 69,
        "y": 69,
        "z_index": 1
    }
],
    "vumeter": true
},
    "result": "ok"
}

```

7.5 POST /output/source/set

功能说明

设置视频源到某一窗口的某个位置上

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
from	body	object	是	来源	待移动的信源信息
» type	body	string	是	来源类型	"source"=信源列表中的流
» stream_id	body	string	是	流ID	流唯一标识, origin和dest必须一致
» stream_name	body	string	是	流名称	流名称
» stream_url	body	string	是	流地址	流地址
» pos_id	body	string	否	位置ID	来源为source时为空串""
» output_id	body	string	是	输出ID	"1"=Output1, "2"=Output2
» layout_id	body	string	否	布局ID	来源为source时为空串""
to	body	object	是	目标	目标位置的布局信息
» type	body	string	是	目标类型	"output"=输出到指定布局窗口
» stream_id	body	string	是	流ID	与from.stream_id一致
» stream_name	body	string	是	流名称	流名称
» stream_url	body	string	是	流地址	流地址
» pos_id	body	integer	是	位置ID	目标布局中的窗口位置, 从1开始
» output_id	body	string	是	输出ID	"1" / "2"
» layout_id	body	integer	是	布局ID	目标布局的唯一标识

Body 请求参数

```
{
  "from": {
    "type": "source",
    "stream_id": "7faa9bc2bcd6962ce869323b0cb161b3",
    "stream_name": "df",
    "stream_url": "udp://172.30.47.230:2555",
    "pos_id": "",
    "output_id": "1",
    "layout_id": ""
  },
  "to": {
    "type": "output",
    "stream_id": "7faa9bc2bcd6962ce869323b0cb161b3",
    "stream_name": "df",
    "stream_url": "udp://172.30.47.230:2555",
    "pos_id": 1,
    "output_id": "1",
  }
}
```

```
"layout_id": 3
}
}
```

返回数据结构

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

7.6 POST /output/source/remove

功能说明

删除窗口上某个位置上绑定的源信息

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» output_id	body	string	是	输出ID	布局ID（1， 2）
» pos_id	body	integer	是	位置ID	位置ID

Body 请求参数

```
{
  "output_id": "1",
  "pos_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

7.7 GET /output/resolution/list

功能说明

获取分辨率（时序）列表

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	返回数据	分辨率列表数据
»» id	string	true	无	分辨率ID	输出分辨率ID
»» name	string	true	无	分辨率名称	输出分辨率名称
»» value	string	true	无	分辨率值	输出分辨率值
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": [
    {
      "id": "1",
      "name": "3840x2160P60(HDMI only)",
      "value": "3840x2160P60(HDMI only)"
    },
    {
      "id": "2",
      "name": "3840x2160P50(HDMI only)",
      "value": "3840x2160P50(HDMI only)"
    },
    {
      "id": "3",
      "name": "3840x2160P30(HDMI only)",
      "value": "3840x2160P30(HDMI only)"
    },
    {
      "id": "4",
      "name": "3840x2160P25(HDMI only)",
      "value": "3840x2160P25(HDMI only)"
    },
  ],
}
```

```
    "id": "5",
    "name": "1920x1080P60",
    "value": "1920x1080P60"
  },
  {
    "id": "6",
    "name": "1920x1080P59.94(SDI only)",
    "value": "1920x1080P59.94(SDI only)"
  },
  {
    "id": "7",
    "name": "1920x1080P50",
    "value": "1920x1080P50"
  },
  {
    "id": "8",
    "name": "1920x1080P30",
    "value": "1920x1080P30"
  },
  {
    "id": "9",
    "name": "1920x1080P29.97(SDI only)",
    "value": "1920x1080P29.97(SDI only)"
  },
  {
    "id": "10",
    "name": "1920x1080P25",
    "value": "1920x1080P25"
  },
  {
    "id": "11",
    "name": "1920x1080P24",
    "value": "1920x1080P24"
  },
  {
    "id": "12",
    "name": "1920x1080P23.98(SDI only)",
    "value": "1920x1080P23.98(SDI only)"
  },
  {
    "id": "13",
    "name": "1920x1080i60",
    "value": "1920x1080i60"
  },
  {
    "id": "14",
    "name": "1920x1080i50",
    "value": "1920x1080i50"
  },
  {
    "id": "15",
    "name": "1920x1080i59.94(SDI only)",
    "value": "1920x1080i59.94(SDI only)"
  }
```

```

    },
    {
      "id": "16",
      "name": "1280x720P60",
      "value": "1280x720P60"
    },
    {
      "id": "17",
      "name": "1280x720P59.94(SDI only)",
      "value": "1280x720P59.94(SDI only)"
    },
    {
      "id": "18",
      "name": "1280x720P50",
      "value": "1280x720P50"
    },
    {
      "id": "19",
      "name": "576i/625i 50Hz(SDI only)",
      "value": "576i/625i 50Hz(SDI only)"
    },
    {
      "id": "20",
      "name": "480i/525i 60Hz(SDI only)",
      "value": "480i/525i 60Hz(SDI only)"
    }
  ],
  "result": "ok"
}

```

7.8 POST /output/resolution/set

功能说明

设置分辨率（时序）列表

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» output_id	body	string	是	输出ID	输出ID
» res_id	body	string	是	分辨率ID	取自 /output/resolution/list 的 id，为 字符串 （如 "14"），不可传整数

Body 请求参数

```
{
  "output_id": "1",
  "res_id": "14"
}
```

返回数据结构

状态码 200

说明：本接口成功时仅返回 {"result":"ok"}，不返回 data。

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	ok

返回示例

```
{
  "result": "ok"
}
```

7.9 GET /output/audiomix/get

功能说明

获取某一窗口的混音音频状态列表

请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	否	输出ID	布局ID
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	无	返回数据	混音音频数据
»» output	[object]	true	无	Output 类型	当前output音频源信息 类型分为两种：output类型根据布局切换变动,preview则是单个output共有
»»» enable	boolean	false	无	启用状态	是否开启

名称	类型	必选	约束	中文名	说明
»»» mute	boolean	false	无	静音	是否静音
»»» name	string	false	无	名称	名称
»»» pos_id	integer	false	无	位置ID	位置ID
»»» stream_id	string	false	无	流ID	流ID
»»» type	string	false	无	类型	类型
»»» volume	integer	false	无	音量	增益dB值
»» preview	[object]	true	无	预览类型	当前preview音频源信息
»»» enable	boolean	false	无	启用状态	是否开启
»»» mute	boolean	false	无	静音	是否静音
»»» name	string	false	无	名称	名称
»»» stream_id	string	false	无	流ID	流ID
»»» type	string	false	无	类型	类型
»»» volume	integer	false	无	音量	增益dB值
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": {
    "output": [
      {
        "stream_id": "1",
        "name": "n60...",
        "enable": false,
        "volume": 0,
        "mute": false,
        "type": "output"
      },
      {
        "stream_id": "2",
        "name": "n50...",
        "enable": false,
        "volume": 0,
        "mute": false,
        "type": "output"
      }
    ]
  }
}
```

```
    ],
    "preview": [
      {
        "stream_id": "1",
        "name": "n50...",
        "enable": false,
        "volume": 0,
        "mute": false,
        "type": "preview"
      }
    ]
  },
  "result": "ok"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	请求成功	Inline

7.10 POST /output/audiomix/set

功能说明

设置某一窗口中的混音音频状态

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» output_id	body	string	是	输出ID	布局ID
» stream_id	body	string	是	流ID	流ID
» enable	body	boolean	是	启用状态	是否开启
» type	body	string	是	类型	类型 支持 output/preview
» volume	body	integer	否	音量	增益dB值（会被读取并生效）
» mute	body	boolean	否	静音	是否静音（会被读取并生效）

Body 请求参数

```
{
  "output_id": "1",
  "stream_id": "895a2714dc8e5197fb0491066be73f22",
  "enable": true,
  "volume": 0,
  "mute": false,
  "type": "output"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

7.11 POST /output/audiomix/remove

功能说明

删除手动添加音频列表中的音频

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» output_id	body	string	是	输出ID	布局ID
» stream_id	body	string	是	流ID	流ID
» type	body	string	是	类型	类型 支持 output/preview

Body 请求参数

```
{
  "output_id": "1",
  "stream_id": "e893d03a94326b981fd3aa276f90183a",
  "type": "preview"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

7.12 POST /output/audiomix/add

功能说明

添加预览中的音频到音频列表中

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» output_id	body	string	是	输出ID	输出ID
» stream_id	body	string	是	流ID	流ID

Body 请求参数

```
{
  "output_id": "1",
  "stream_id": "895a2714dc8e5197fb0491066be73f22"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

7.13 POST /ptz/set

功能说明

设置流的PTZ（目前只支持NDI）

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» output_id	body	string	是	输出ID	输出ID
» pos_id	body	integer	是	位置ID	位置ID
» type	body	string	是	类型	类型，单个字符串（见下方 type 取值表），不可传数组
» data	body	object	是	请求数据	具体值；实际只需携带当前 type 所需的字段
»» zoom_value	body	number	否	变焦值	0.0 (zoomed in) ... 1.0 (zoomed out)
»» zoom_speed	body	number	否	变焦速度	-1.0 (zoom outwards) ... +1.0 (zoom inwards)
»» pan_value	body	number	否	水平转动值	-1.0 (left) ... 0.0 (centered) ... +1.0 (right)
»» tilt_value	body	number	否	垂直转动值	-1.0 (bottom) ... 0.0 (centered) ... +1.0 (top)
»» pan_speed	body	number	否	水平转动速度	-1.0 (moving right) ... 0.0 (stopped) ... +1.0 (moving left)
»» tilt_speed	body	number	否	垂直转动速度	-1.0 (down) ... 0.0 (stopped) ... +1.0 (moving up)
»» preset_no	body	integer	否	预设编号	0 ... 99
»» speed	body	number	否	速度	0.0(as slow as possible) ... 1.0(as fast as possible)
»» focus_value	body	number	否	对焦值	0.0 (focused to infinity) ... 1.0 (focused as close as possible)
»» focus_speed	body	number	否	对焦速度	-1.0 (focus outwards) ... +1.0 (focus inwards)
»» red	body	number	否	红色增益	0.0(not red) ... 1.0(very red)

名称	位置	类型	必选	中文名	说明
»» blue	body	number	否	蓝色增益	0.0(not blue) ... 1.0(very blue)
»» exposure_level	body	number	否	曝光等级	0.0(dark) ... 1.0(light)

type字段支持参数如下

类型名称	功能说明
ptz_zoom	控制 PTZ 的变焦操作，需要data对象中包含zoom_value字段，设置变焦值
ptz_zoom_speed	控制 PTZ 的变焦速度，需要data对象中包含zoom_speed字段，设置变焦速度值
ptz_pan_tilt	控制 PTZ 的水平（pan）和垂直（tilt）转动，需要data对象中包含pan_value和tilt_value字段，分别设置水平和垂直转动的值
ptz_pan_tilt_speed	控制 PTZ 的水平和垂直转动速度，需要data对象中包含pan_speed和tilt_speed字段，分别设置水平和垂直转动速度值
ptz_store_preset	存储 PTZ 预设位置，需要data对象中包含preset_no字段，设置预设位置编号
ptz_recall_preset	调用 PTZ 预设位置，需要data对象中包含preset_no和speed字段，分别设置预设位置编号和调用速度
ptz_auto_focus	启用自动对焦功能，无需参数
ptz_focus	控制 PTZ 的对焦操作，需要data对象中包含focus_value字段，设置对焦值
ptz_focus_speed	控制 PTZ 的对焦速度，需要data对象中包含focus_speed字段，设置对焦速度值
ptz_white_balance_auto	启用自动白平衡，无需参数
ptz_white_balance_indoor	设置室内白平衡模式，无需参数
ptz_white_balance_outdoor	设置室外白平衡模式，无需参数
ptz_white_balance_one-shot	执行一次性白平衡调整，无需参数
ptz_white_balance_manual	手动设置白平衡，需要data对象中包含 red（红色增益）和 blue（蓝色增益）
ptz_exposure_auto	启用自动曝光，无需参数
ptz_exposure_manual	手动设置曝光等级，需要data对象中包含 exposure_level
ptz_is_supported	查询设备是否支持PTZ功能，无需参数。 支持则返回 {"result":"ok"}， 不支持则返回 {"result":"error","msg":"508"}（无布尔字段）

速度控制类必须重置> 示例：设置PTZ需要分成两步：1、设置目标类型以及目标参数step1:{"type": "ptz_focus_speed","data": {"focus_speed": 0.5},"pos_id": 1,"output_id": "1"}2、设置目标类型以及参数归原step2: {"type": "ptz_focus_speed","data": {"focus_speed": 0},"pos_id": 1,"output_id": "1"}
绝对位置类接口无需重置> 示例：直接设置即可1、设置目标类型以及目标参数step1:{"type": "ptz_recall_preset","data": {"preset_no": 1,"speed": 0.5},"pos_id": 1,"output_id": "1"}

Body 请求参数

```
{
  "output_id": "1",
  "pos_id": 1,
  "type": "ptz_zoom",
  "data": {
    "zoom_value": 0.5,
    "zoom_speed": -1.0,
    "pan_value": -1.0,
    "tilt_value": -1.0,
    "pan_speed": -1.0,
    "tilt_speed": -1.0,
    "preset_no": 0,
    "speed": 0.0,
    "focus_value": 0.0,
    "focus_speed": -1.0,
    "red": 0.0,
    "blue": 0.0,
    "exposure_level": 0.0
  }
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

7.14 GET /output/audio/sound

功能说明

获取音频音柱

请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	是	输出ID	输出ID
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	返回数据	音频音柱数据
»» dB	[number]	true	无	音频dB值	音频dB值（浮点数组）
»» stream_id	string	true	无	流ID	流ID
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": [
    {
      "dB": [
        0,
        0,
        0,
        0
      ],
      "stream_id": "e893d03a94326b981fd3aa276f90183a"
    },
    {
      "dB": [
        0,
        0,
        0,
        0
      ],
      "stream_id": "e893d03a94326b981fd3aa276f90183a"
    }
  ],
  "result": "ok"
}
```

7.15 POST /output/border/set

功能说明

设置布局边框

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» output_id	body	string	是	输出ID	输出ID
» enable	body	boolean	是	启用状态	是否开启
» color	body	string	是	边框颜色	边框颜色

Body 请求参数

```
{
  "output_id": "1",
  "enable": true,
  "color": "#FF0000"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

7.16 GET /output/border/get

功能说明

获取布局边框配置

请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	integer	是	输出ID	输出ID
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	无	返回数据	边框配置数据
»» bottom_width	integer	true	无	下边框宽度	边框宽度
»» color	string	true	无	边框颜色	边框颜色
»» enable	boolean	true	无	启用状态	是否开启
»» hex_color	integer	true	无	十六进制颜色值	边框颜色的十六进制数值
»» left_width	integer	true	无	左边框宽度	边框宽度
»» right_width	integer	true	无	右边框宽度	边框宽度
»» top_width	integer	true	无	上边框宽度	边框宽度
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": {
    "bottom_width": 2,
    "color": "#FF0000",
    "enable": true,
    "hex_color": 16711680,
    "left_width": 2,
    "right_width": 2,
    "top_width": 2
  },
  "result": "ok"
}
```

7.17 POST /output/background/set

功能说明

设置布局背景

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» output_id	body	string	是	输出ID	输出ID
» enable	body	boolean	是	启用状态	是否开启
» background	body	string	是	背景颜色	背景颜色

Body 请求参数

```
{
  "output_id": "1",
  "enable": true,
  "background": "#000000"
}
```

返回数据结构

返回示例

```
{
  "result": "ok"
}
```

7.18 GET /output/background/get

功能说明

获取布局背景配置

请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	string	否	输出ID	输出ID
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	无	返回数据	背景配置数据
»» background	string	true	无	背景颜色	背景颜色
»» enable	boolean	true	无	启用状态	是否开启
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": {
    "background": "",
    "enable": true
  },
  "result": "ok"
}
```

7.19 POST /output/audio/check

功能说明

获取音频是否可选状态

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» id	body	string	是	音频设备ID	音频设备ID

Body 请求参数

```
{
  "id": ""
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	boolean	true	无	返回数据	是否允许打开
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": true,
  "result": "ok"
}
```

7.20 POST /output/ts/program/set

功能说明

多节目多音轨切换

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» stream_id	body	string	是	视频源ID	视频源ID
» selected_id	body	integer	是	节目ID	节目ID

Body 请求参数

```
{
  "stream_id": "7faa9bc2bcd6962ce869323b0cb161b3",
  "selected_id": 3
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

7.21 GET /output/ts/program/get

功能说明

获取多节目列表

请求参数

名称	位置	类型	必选	中文名	说明
stream_id	query	string	是	视频源ID	视频源的唯一标识
app	header	string	是	鉴权令牌	应用标识

Body 请求参数

```
{}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	无	返回数据	节目数据
»» selected_id	integer	true	无	选中节目ID	当前选中的节目/音轨ID
»» selection	[array]	true	无	选中项	[节目名称, 节目/音轨ID]
»»» [0]	string	true	无	节目名称	选中项的名称
»»» [1]	integer	true	无	选中ID	选中项的ID
»» supported	boolean	true	无	是否支持	是否支持多节目选择
»» tree	[object]	true	无	节目树	节目列表树形结构
»»» children	[object]	true	无	音频子节点	节目下的音频流子节点列表
»»»» id	integer	true	无	音频ID	音频流ID
»»»» name	string	true	无	音频名称	音频流显示名称
»»»» pid	integer	true	无	PID	音频流PID值
»»» id	string	true	无	节目ID	节目标识符
»»» name	string	true	无	节目名称	节目显示名称
» result	string	true	无	返回结果	请求结果状态 ("ok")

返回示例

```
{
  "data": {
    "selected_id": 3,
    "selection": [
      "program_2",
      3
    ],
    "supported": true,
    "tree": [
      {
        "children": [
          {
            "id": 1,
            "name": "Audio (stream 1)",
```

```
        "pid": 257
      }
    ],
    "id": "program_1",
    "name": "DSP Uy Premium"
  },
  {
    "children": [
      {
        "id": 3,
        "name": "Audio (stream 3)",
        "pid": 258
      }
    ],
    "id": "program_2",
    "name": "DSP Uy Basico"
  }
]
},
"result": "ok"
}
```

7.22 GET /output/encode/get

功能说明

获取编码参数

请求参数

名称	位置	类型	必选	中文名	说明
output_id	query	integer	是	输出ID	1=Output1, 2=Output2
app	header	string	是	鉴权令牌	应用标识

返回数据结构

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态
» data	object	true	无	返回数据	编码参数数据
»» enable	boolean	true	无	编码启用	是否启用编码
»» main	object	true	无	主码流参数	主码流参数
»»» codec	string	true	无	编码类型	"H264" / "H265"
»»» profile	integer	true	无	编码档次	0=High, 1=Main, 2=Baseline

名称	类型	必选	约束	中文名	说明
»»» pic_width	integer	true	无	编码宽度	set_resolution_default=true时为1, 否则为实际像素
»»» pic_height	integer	true	无	编码高度	set_resolution_default=true时为1, 否则为实际像素
»»» set_resolution_default	boolean	true	无	使用默认分辨率	true时宽高使用输出分辨率
»»» grey	boolean	true	无	灰度	true=灰度, false=彩色
»»» fps	string	true	无	帧率	"auto"=自动, "half"=半帧, 数字=固定帧率
»»» mode	string	true	无	码率模式	"CBR"=固定码率, "VBR"=可变码率
»»» bitrate	integer	true	无	码率	kbps; 32~50000
»»» max_bitrate	integer	true	无	最大码率	kbps; VBR模式下生效
»»» gop	integer	true	无	GOP大小	1/30/60/90/120/150/180/自定义
»»» deinterlace_mode	string	true	无	去隔行	"weave"=默认, "line_doubling"=场模式
»»» code_frame	integer	true	无	参考帧	0=单帧, 非0=多帧
»»» out_mixer_fps	integer	true	无	混音输出帧率	Hz
»» sub	object	true	无	子码流参数	子码流参数
»»» pic_width	integer	true	无	宽度	固定640
»»» pic_height	integer	true	无	高度	固定360
»»» fps	integer	true	无	帧率	固定30
»»» bitrate	integer	true	无	码率	主码率/2, kbps

状态码 200

返回示例

```
{
  "result": "ok",
  "data": {
    "enable": false,
    "main": {
      "codec": "H264",
      "profile": 0,
      "pic_width": 1,
```

```

    "pic_height": 1,
    "set_resolution_default": true,
    "grey": false,
    "fps": "auto",
    "mode": "VBR",
    "bitrate": 6000,
    "max_bitrate": 20000,
    "gop": 60,
    "deinterlace_mode": "weave",
    "code_frame": 0,
    "out_mixer_fps": 60
  },
  "sub": {
    "pic_width": 640,
    "pic_height": 360,
    "fps": 30,
    "bitrate": 3000
  }
}

```

7.23 POST /output/encode/set

功能说明

设置编码参数

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
output_id	body	integer	是	输出ID	1或2
main	body	object	是	主码流参数	主码流参数
» codec	body	string	否	编码类型	"H264" / "H265"
» grey	body	boolean	否	灰度	true=灰度, false=彩色
» profile	body	integer	否	编码档次	0=High, 1=Main, 2=Baseline
» pic_width	body	integer	否	编码宽度	传1时使用输出分辨率
» pic_height	body	integer	否	编码高度	传1时使用输出分辨率
» mode	body	string	否	码率模式	"CBR" / "VBR"
» gop	body	integer	否	GOP大小	1/30/60/90/120/150/180
» fps	body	string	否	帧率	"auto" / "half" / 数字字符串
» bitrate	body	integer	否	码率	kbps; 32~50000

名称	位置	类型	必选	中文名	说明
» max_bitrate	body	integer	否	最大码率	kbps; VBR模式下生效
» code_frame	body	integer	否	参考帧	0=单帧, 非0=多帧
» deinterlace_mode	body	string	否	去隔行	"default" / "line_doubling"

Body 请求参数

```
{
  "output_id": 1,
  "main": {
    "codec": "H265",
    "grey": false,
    "profile": 0,
    "pic_width": 1,
    "pic_height": 1,
    "mode": "VBR",
    "gop": 60,
    "fps": "auto",
    "bitrate": 6000,
    "max_bitrate": 20000,
    "code_frame": 0,
    "deinterlace_mode": "default"
  }
}
```

返回数据结构

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

状态码 200

返回示例

```
{
  "result": "ok"
}
```

7.24 POST /output/vumeter/set

功能说明

设置音柱开关

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
enable	body	boolean	是	启用音柱	true=开启, false=关闭
output_id	body	string	是	输出ID	"1"=Output1, "2"=Output2
pos_id	body	integer	是	位置ID	传-1设置全部位置

Body 请求参数

```
{
  "enable": false,
  "output_id": "1",
  "pos_id": 2
}
```

返回数据结构

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

状态码 200

返回示例

```
{
  "result": "ok"
}
```

7.25 POST /output/mute/set

功能说明

设置静音

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
output_id	body	integer	是	输出ID	1或2
pos_id	body	integer	是	位置ID	视频位置ID
mute	body	boolean	是	静音	true=静音, false=取消静音

Body 请求参数

```
{
  "output_id": 1,
  "pos_id": 0,
  "mute": true
}
```

返回数据结构

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

状态码 200

返回示例

```
{
  "result": "ok"
}
```

8. GATE模块

GATE（网关推流）模块用于管理设备将本地视频流转推到外部服务器的配置。支持多种推流协议（RTMP / RTSP / RTP / TS / SRT / HLS / NDI_HX 等），可对每条推流进行独立启用/禁用、绑定源视频流、批量操作以及多路输出等管理。

8.1 GET /gate/stream/list

功能说明

获取媒体网关流列表信息

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	应用标识

返回数据结构

状态码 200

公共字段（所有网关流均返回）

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	流列表	网关流对象数组
»» id	string	true	无	流ID	网关列表中的唯一ID
»» index	integer	true	无	索引	网关列表中的流索引

名称	类型	必选	约束	中文名	说明
»» name	string	true	无	名称	网关流名称
»» protocol	string	true	无	协议类型	流协议类型
»» is_enable	boolean	true	无	启用状态	是否已开启推流
»» src_url	string	true	无	源地址	拉流源地址
»» stream_id	string	true	无	流ID	关联的源流ID
»» stream_name	string	true	无	流名称	源流名称
»» error_code	integer	true	无	错误码	推流错误码，0 表示正常
»» error_msg	string	true	无	错误描述	推流错误描述
»» output_state	string	true	无	输出状态	推流输出状态
»» video_stream_info	object	true	无	视频信息	视频流信息
»»» codec_id	string	true	无	编码类型	视频编码类型
»»» format	string	true	无	帧类型	p帧或i帧
»»» fps	integer	true	无	帧率	视频帧率
»»» height	integer	true	无	高度	视频高度
»»» width	integer	true	无	宽度	视频宽度
»» audio_stream_info	object	true	无	音频信息	音频流信息
»»» bits	integer	true	无	位深	音频位深
»»» channels	integer	true	无	声道数	音频声道数
»»» codec_id	string	true	无	编码类型	音频编码类型
»»» sample_rate	integer	true	无	采样率	音频采样率
»» net_statics	object	true	无	网络统计	网络传输统计
»»» down_kbps	integer	true	无	下行速率	下行速率（kbps）
»»» up_kbps	integer	true	无	上行速率	上行速率（kbps）
»» ts_program_num	integer	true	无	TS节目号	TS 流节目号，无则为 -1

 以下为各协议类型专属字段。

RTSP 协议返回字段

名称	类型	必选	约束	中文名	说明
»» auth	boolean	true	无	鉴权启用	是否启用身份验证
»» user	string	true	无	用户名	RTSP 鉴权用户名
»» pwd	string	true	无	密码	RTSP 鉴权密码
»» session	string	true	无	会话名称	RTSP 会话名称/通道号
»» dest_url	string	true	无	目标地址	目标推送地址
»» port	integer	true	无	端口	RTSP 流传输端口
»» http_tunnel_port	integer	true	无	HTTP隧道端口	HTTP隧道端口号
»» multicast_enable	boolean	true	无	组播启用	是否启用组播
»» multicast_addr	string	true	无	组播地址	组播地址
»» multicast_port_min	integer	true	无	组播端口最小值	组播端口范围下限
»» multicast_port_max	integer	true	无	组播端口最大值	组播端口范围上限
»» multicast_ttl	integer	true	无	组播TTL	组播数据包生存时间

RTP 协议返回字段

名称	类型	必选	约束	中文名	说明
»» dest_url	string	true	无	目标地址	目标推送地址
»» port	integer	true	无	端口	RTP 传输端口
»» load_type	string	false	无	负载类型	RTP 负载类型（如 ts）
»» ttl	integer	true	无	TTL	RTP 数据包生存时间

TS 协议返回字段

名称	类型	必选	约束	中文名	说明
»» dest_url	string	true	无	目标地址	目标推送地址
»» port	integer	true	无	端口	TS 传输端口
»» ts_start_pid	integer	true	无	起始PID	媒体数据起始PID
»» ts_pmt_start_pid	integer	true	无	PMT起始PID	节目映射表起始PID
»» ts_service_name	string	true	无	服务名称	TS 服务名称
»» ts_service_provider	string	true	无	服务提供商	TS 服务提供商名称

名称	类型	必选	约束	中文名	说明
»» ts_tables_version	integer	true	无	表版本号	TS 表的版本号
»» ts_transport_stream_id	integer	true	无	传输流ID	TS 传输流ID
»» ts_pcr_period	integer	true	无	PCR时钟周期	PCR 时钟周期（ms）
»» ts_null_multiple	integer	true	无	空包倍数	TS 空包填充倍数
»» bind_port	integer	false	无	绑定端口	TS 绑定端口，0 表示未绑定
»» ts_pts_pcr_delay	integer	false	无	PTS/PCR延迟	PTS/PCR 延迟（ms）

SRT 协议返回字段

名称	类型	必选	约束	中文名	说明
»» connection_mode	string	false	无	连接模式	Listener、Caller 或 Rendezvous
»» encryption	string	false	无	加密状态	0=不加密，1=启用加密
»» latency	integer	false	无	传输延迟	SRT 传输延迟（ms）
»» listener_port	integer	false	无	监听端口	SRT 监听端口
»» passphrase	string	false	无	加密密码	SRT 加密密码
»» srt_stream_id	string	false	无	SRT流ID	SRT 流唯一标识

HLS 协议返回字段

名称	类型	必选	约束	中文名	说明
»» mode	string	false	无	运行模式	HLS 运行模式（如 server）
»» segment_time	integer	false	无	分片时长	HLS 分片时长（秒）
»» max_segments	integer	false	无	最大分片数	HLS 保留分片数
»» media_playlist_url	string	false	无	播放列表地址	HLS 播放列表（m3u8）

NDI HX 协议返回字段

名称	类型	必选	约束	中文名	说明
»» channel_name	string	false	无	通道名称	NDI 通道名称
»» connection	string	false	无	连接方式	NDI 连接方式
»» discovery_server	string	false	无	发现服务器	NDI 发现服务器地址

名称	类型	必选	约束	中文名	说明
»» group	string	false	无	发现组	NDI 发现组
»» netmask	string	false	无	子网掩码	NDI 流子网掩码
»» netprefix	string	false	无	网络前缀	NDI 流网络前缀

RTMP 协议返回字段

名称	类型	必选	约束	中文名	说明
»» address	string	true	无	推送地址	RTMP 推送地址
»» dest_url	string	true	无	目标地址	目标推送地址
»» user	string	true	无	用户名	RTMP 鉴权用户名
»» pwd	string	true	无	密码	RTMP 鉴权密码
»» password	string	false	无	密码	推流密码
»» conn_intv	integer	false	无	重连间隔	重连间隔（秒）
»» conn_timeout	integer	false	无	连接超时	连接超时（秒）
»» old_rtmp	boolean	false	无	旧版RTMP	是否启用旧版 RTMP

全局返回字段

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果, "ok" 表示成功

返回示例

```
200 Response
{
  "data": [{
    "audio_stream_info": {
      "bits": 0,
      "channels": 2,
      "codec_id": "AAC",
      "sample_rate": 48000
    },
    "auth": false,
    "dest_url": "",
    "http_tunnel_port": 8554,
    "id": "f29fe5e3c92814a81def018fd6c980df",
    "index": 34,
    "is_enable": false,
```

```
{
  "multicast_addr": "224.0.1.0",
  "multicast_enable": false,
  "multicast_port_max": 31004,
  "multicast_port_min": 31000,
  "multicast_ttl": 127,
  "name": "RS96",
  "net_statics": {
    "down_kbps": 0,
    "up_kbps": 0
  },
  "output_state": "inactive",
  "port": 596,
  "protocol": "rtsp",
  "pwd": "",
  "session": "ch96",
  "src_url": "rtsp://172.30.48.227:596/ch96",
  "stream_id": "c9c7339a070aa9906b263994e5cbaf4e",
  "stream_name": "",
  "user": "",
  "video_stream_info": {
    "codec_id": "H.264",
    "format": "p",
    "fps": 60,
    "height": 1080,
    "width": 1920
  }
},
],
"result": "ok"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	请求成功	Inline

8.2 POST /gate/stream/add

功能说明

添加媒体网关流

Body 请求参数

RTMP 示例（其余协议见下方各协议参数表）：

```
{
  "name": "推流1",
  "protocol": "rtmp",
  "is_enable": true,
  "address": "rtmp://192.168.1.100/live/stream",
  "user": "",
  "password": "",
  "conn_intv": 3,
  "conn_timeout": 10,
  "old_rtmp": false
}
```

公共请求参数（所有协议）

`protocol` 决定本条网关流的类型，取值：`rtmp` / `rtsp` / `rtp` / `ts` / `srt` / `hls` / `ndi_hx`（`ndi_full` 保留）。不同 `protocol` 需要携带**不同的**协议专属字段（见下方各协议表）；缺少某协议的必填字段会返回 `1`（参数错误）导致添加失败。

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» name	body	string	是	网关流名称	网关流名称（非空）
» protocol	body	string	是	协议类型	协议类型，见上
» is_enable	body	boolean	是	开启状态	是否开启推流
» stream_id	body	string	否	流ID	绑定的源流ID；也可后续用 /gate/stream/bind_src 绑定
» stream_name	body	string	否	流名称	流名称

协议专属参数

说明：以下各表列出**除公共参数外**每种协议实际会被校验的字段。`ts_*` 系列为 TS 封装参数，带范围的字段超范围会被拒绝。

`protocol = rtmp`

名称	类型	必选	中文名	说明
» address	string	是	推送地址	RTMP 推送地址（非空）
» conn_intv	integer	是	重连间隔	重连间隔（秒），需 > 0

名称	类型	必选	中文名	说明
» conn_timeout	integer	是	连接超时	连接超时（秒），需 > 0
» old_rtmp	boolean	是	旧版RTMP	是否启用旧 RTMP 版本
» user	string	否	用户名	用户名
» password	string	否	密码	密码

protocol = rtsp

名称	类型	必选	中文名	说明
» session	string	是	会话名称	RTSP 会话名
» port	integer	是	端口	RTSP 服务端口
» http_tunnel_port	integer	是	HTTP隧道端口	HTTP 隧道端口
» auth	boolean	是	是否鉴权	是否启用鉴权
» user	string	auth=true 时必填	用户名	用户名
» pwd	string	auth=true 时必填	密码	密码
» multicast_enable	boolean	是	组播启用	是否启用组播
» multicast_addr	string	是	组播地址	组播地址
» multicast_ttl	integer	是	组播TTL	组播 TTL
» multicast_port_min	integer	是	组播端口下限	组播端口范围下限
» multicast_port_max	integer	是	组播端口上限	组播端口范围上限

protocol = rtp (RTP over UDP)

名称	类型	必选	中文名	说明
» address	string	是	目标地址	目标地址
» port	integer	是	目标端口	目标端口
» load_type	integer	是	负载类型	负载类型
» ttl	integer	是	TTL	TTL
» ts_service_name	string	是	服务名称	TS 服务名
» ts_service_provider	string	是	服务提供商	TS 服务提供商
» ts_pcr_period	integer	是	PCR周期	PCR 周期，1-100
» ts_transport_stream_id	integer	是	TS ID	TS ID，1-65535

名称	类型	必选	中文名	说明
» ts_start_pid	integer	是	起始PID	起始 PID, 32-3840
» ts_pmt_start_pid	integer	是	PMT起始PID	PMT 起始 PID, 16-7936
» ts_tables_version	integer	是	表版本	表版本, 0-31

protocol = ts (RTP over TS / UDP)

名称	类型	必选	中文名	说明
» address	string	是	目标地址	目标地址
» port	integer	是	目标端口	目标端口
» bind_port	integer	是	绑定端口	本地绑定端口, 0-65535
» ttl	integer	是	TTL	TTL
» ts_pts_pcr_delay	integer	是	PTS/PCR延迟	PTS/PCR 延迟
» ts_service_name	string	是	服务名称	TS 服务名
» ts_service_provider	string	是	服务提供商	TS 服务提供商
» ts_pcr_period	integer	是	PCR周期	PCR 周期, 1-100
» ts_transport_stream_id	integer	是	TS ID	TS ID
» ts_start_pid	integer	是	起始PID	起始 PID
» ts_pmt_start_pid	integer	是	PMT起始PID	PMT 起始 PID
» ts_tables_version	integer	是	表版本	表版本

protocol = srt

名称	类型	必选	中文名	说明
» connection_mode	string	是	连接模式	连接模式 (Caller/Listener/Rendezvous)
» address	string	非 Listener 时必填	目标地址	目标地址
» listener_port	integer	是	监听端口	监听/连接端口, 1025-65535
» latency	integer	是	延迟	延迟(ms), 0-5000
» encryption	string	是	加密方式	加密方式, "0" 表示不加密
» passphrase	string	encryption≠"0" 时必填	密钥	密钥, 10-79 字符
» ts_service_name	string	是	服务名称	TS 服务名

名称	类型	必选	中文名	说明
» ts_service_provider	string	是	服务提供商	TS 服务提供商
» ts_pcr_period	integer	是	PCR周期	PCR 周期
» ts_start_pid	integer	是	起始PID	起始 PID
» ts_pmt_start_pid	integer	是	PMT起始PID	PMT 起始 PID
» ts_tables_version	integer	是	表版本	表版本

protocol = hls

名称	类型	必选	中文名	说明
» mode	string	是	模式	模式, "server" 时需再填 session
» session	string	mode="server" 时必填	会话ID	会话 ID (重复返回 615)
» segment_time	integer	是	分片时长	分片时长, 需 > 0
» max_segments	integer	是	最大分片数	最大分片数, 需 > 0

protocol = ndi_hx

名称	类型	必选	中文名	说明
» channel_name	string	是	NDI通道名	NDI 通道名 (重复返回 614)
» ttl	integer	是	TTL	TTL, 需 > 0
» connection	string	是	连接方式	连接方式, "multicast" 时需再填 netmask/netprefix
» netmask	string	connection="multicast" 时必填	子网掩码	组播子网掩码
» netprefix	string	connection="multicast" 时必填	网络前缀	组播网络前缀

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
200 Response
{
  "result": "ok"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	请求成功	Inline

8.3 POST /gate/stream/delete

功能说明

删除媒体网关流

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» ids	body	[object]	是	流数组	流数组
»» id	body	string	是	流ID	网关列表中的唯一ID

Body 请求参数

```
{
  "ids": [
    {
      "id": "a808357e97d12ecce12f6e0537654a69"
    },
    {
      "id": "3cce25843301e212ce1dab5ebdc0ad91"
    },
    {
      "id": "f8e9900cc04a378eaa2b92d5dd4edf90"
    }
  ]
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
200 Response
{
  "result": "ok"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	请求成功	Inline

8.4 POST /gate/stream/update

功能说明

修改媒体网关流信息

请求参数

说明：update 采用**增量合并**——按 `id` 定位记录后，将本次提交的字段合并进原记录。因此**只有 `id` 是必填**，其余字段只提交需要修改的即可；未提交的字段保持原值。`output_state` 是服务端计算的推流状态，非有效入参，无需提交。修改协议专属字段时，字段含义与取值范围同 `/gate/stream/add` 各协议表。

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» id	body	string	是	流ID	网关列表中的唯一ID（定位待修改记录）
» name	body	string	否	网关流名称	网关流名称
» stream_id	body	string	否	流ID	流ID
» stream_name	body	string	否	流名称	流名称
» protocol	body	string	否	协议类型	协议类型
» is_enable	body	boolean	否	开启状态	是否开启推流
» password	body	string	否	密码	密码
» address	body	string	否	推送地址	RTMP推送地址
» user	body	string	否	用户名	用户名
» conn_intv	body	integer	否	重连间隔	重连间隔（秒）

名称	位置	类型	必选	中文名	说明
» conn_timeout	body	integer	否	连接超时	连接超时（秒）
» old_rtmp	body	boolean	否	旧版RTMP	启用旧RTMP版本
» (其他协议字段)	body	-	否	其他字段	见 /gate/stream/add 各协议参数表

Body 请求参数

```
{
  "address": "rtmp://172.30.227:1935/live/1",
  "user": "admin",
  "password": "Admin123",
  "conn_timeout": 15,
  "conn_intv": 3,
  "old_rtmp": false,
  "id": "82c83493f98abf452b7b6edccca98a52",
  "name": "rtmp0",
  "is_enable": false,
  "protocol": "rtmp",
  "stream_id": "c9c7339a070aa9906b263994e5cbaf4e"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
200 Response
{
  "result": "ok"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	请求成功	Inline

8.5 POST /gate/stream/bind_src

功能说明

绑定媒体网关推流源

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» stream_id	body	string	是	流ID	流ID
» id	body	string	是	网关流ID	网关列表中的唯一ID

Body 请求参数

```
{
  "id": "b316084c326e7166e2f9af51e4d2b9dc",
  "stream_id": "1feb0d043a38db99d212199942785289"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
200 Response
{
  "result": "ok"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	请求成功	Inline

8.6 POST /gate/stream/batch_enable

功能说明

批量开启媒体网关推流

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» ids	body	[object]	是	流ID数组	网关流ID列表
»» id	body	string	是	流ID	网关列表中的唯一ID
» enable	body	boolean	是	开启状态	是否开启网关推流

Body 请求参数

```
{
  "ids": [
    {
      "id": "afab531cee66302083f4966286652f56"
    }
  ],
  "enable": true
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
200 Response
{
  "result": "ok"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	请求成功	Inline

8.7 GET /gate/stream/multi_out_query

功能说明

查询OUTPUT1/2编码输出预览的开启状态

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	无	返回数据	编码输出预览配置
»» enable	integer	true	无	当前窗口	切换OUTPUT1/2编码输出预览窗口
»» skip_switch_confirm	boolean	true	无	跳过确认	是否跳过切换确认弹窗
» result	string	true	无	返回结果	调用结果

返回示例

```
200 Response
{
  "data": {
    "enable": 1,
    "skip_switch_confirm": false
  },
  "result": "ok"
}
```

返回结果

状态码	状态码含义	说明	数据模型
200	OK	请求成功	Inline

8.8 POST /gate/stream/multi_out_enable

功能说明

切换OUTPUT1/2编码输出预览窗口

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	否	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» enable	body	integer	是	目标窗口	切换OUTPUT1/2编码输出预览窗口
» skip_switch_confirm	body	boolean	否	跳过确认	是否跳过确认弹窗；不传则维持原值（兼容旧版本）

Body 请求参数

```
{
  "enable": 2,
  "skip_switch_confirm": false
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

9. NETWORK模块

NETWORK（网络）模块用于管理设备的网络配置。包括网口 IP 地址与 DHCP 设置、DNS 服务器配置、HTTP 服务端口、静态路由表的管理（添加/删除/修改/启用）以及网络连通性诊断（Ping）等功能。

9.1 GET /network/get

功能说明

获取网卡列表

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	返回数据	网卡列表数据
»» device	string	true	无	网口名称	网口名称
»» dns	string	true	无	DNS	DNS服务器地址
»» dynamic	string	true	无	DHCP状态	是否DHCP动态分配
»» enable	integer	true	无	启用状态	是否开启
»» enable2	boolean	true	无	维护地址启用	维护地址是否开启
»» gw	string	true	无	网关	网关地址
»» gw2	string	true	无	维护地址网关	维护地址网关
»» ip	string	true	无	IP地址	IP地址
»» ip2	string	true	无	维护地址IP	维护地址IP
»» mac	string	true	无	MAC地址	MAC地址
»» mask	string	true	无	子网掩码	子网掩码
»» mask2	string	true	无	维护地址子网掩码	维护地址子网掩码
»» state	string	true	无	连接状态	是否连接状态
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": [
    {
      "device": "eth0",
      "dns": "",
      "dynamic": "n",
      "enable": 1,
      "enable2": true,
      "gw": "192.168.36.254",
      "gw2": "",
      "ip": "192.168.36.6",
      "ip2": "192.168.1.163",
      "mac": "18:C3:F4:25:66:66",
      "mask": "255.255.255.0",
      "mask2": "255.255.255.0",
      "state": "up"
    },
    {
      "device": "eth1",
      "dns": "",
      "dynamic": "n",
```

```
        "enable": 1,
        "enable2": true,
        "gw": "192.168.36.254",
        "gw2": "",
        "ip": "192.168.36.16",
        "ip2": "192.168.2.161",
        "mac": "18:C3:F4:25:66:67",
        "mask": "255.255.255.0",
        "mask2": "255.255.255.0",
        "state": "down"
    }
],
"result": "ok"
}
```

9.2 POST /network/routing/add

功能说明

添加路由

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» device	body	string	是	网口名称	网口名称
» ip	body	string	是	路由IP	路由IP
» netmask	body	string	是	子网掩码	子网掩码
» gw	body	string	是	网关	网关
» mtu	body	integer	是	MTU	MTU
» metric	body	integer	是	Metric	Metric

Body 请求参数

```
{
  "device": "eth1",
  "ip": "192.168.1.1",
  "netmask": "255.255.255.255",
  "gw": "192.168.36.6",
  "mtu": 1500,
  "metric": 0
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

9.3 POST /network/http/set

功能说明

设置HTTP/HTTPS端口

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» http	body	boolean	是	HTTP启用	是否开启HTTP
» httpPort	body	integer	是	HTTP端口	HTTP端口
» https	body	boolean	是	HTTPS启用	是否开启HTTPS
» httpsPort	body	integer	是	HTTPS端口	HTTPS端口
» authEnable	body	boolean	否	鉴权启用	是否开启鉴权

说明：端口若与系统保留端口冲突（99/100/102/6665/6666、RTMP、WebSocket 等）会返回错误 `{"result":"error","msg":"405"}`（ERR_PORT_ALREADY_OCCUPIED）。

Body 请求参数

```
{
  "http": true,
  "httpPort": 80,
  "https": true,
  "httpsPort": 443
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

9.4 GET /network/http/get

功能说明

获取HTTP/HTTPS端口

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	无	返回数据	HTTP/HTTPS端口数据
»» http	boolean	true	无	HTTP启用	是否开启HTTP
»» httpPort	integer	true	无	HTTP端口	HTTP端口
»» https	boolean	true	无	HTTPS启用	是否开启HTTPS
»» httpsPort	integer	true	无	HTTPS端口	HTTPS端口
»» authEnable	boolean	true	无	鉴权启用	是否开启鉴权（恒定返回）
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "data": {
    "http": true,
    "httpPort": 80,
    "https": true,
    "httpsPort": 443,
    "authEnable": false
  },
  "result": "ok"
}
```

9.5 POST /network/ping

功能说明

查看是否能ping通指定地址

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» ip	body	string	是	目标IP	目标地址

Body 请求参数

```
{
  "ip": "192.168.36.124"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true	无	返回数据	Ping结果数据
»» lossRate	integer	true	无	丢包率	丢包率
»» reachable	boolean	true	无	是否可达	是否可达；成功返回时恒为 true
» result	string	true	无	返回结果	请求结果状态

说明：仅当可达时返回上述成功结构（`reachable` 恒为 `true`）；**不可达时返回错误**
`{"result":"error","msg":"400"}`（`ERR_IP_UNREACHABLE`），不会返回 `reachable:false`。

返回示例

```
{
  "data": {
    "lossRate": 0,
    "reachable": true
  },
  "result": "ok"
}
```

9.6 POST /network/routing/set

功能说明

设置路由状态（是否启用）

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» id	body	string	是	路由ID	唯一值
» status	body	integer	是	启用状态	是否开启 1为开启 0为关闭

Body 请求参数

```
{
  "id": "c26163a66109b6c1a717e7be658ab071",
  "status": 0
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

9.7 POST /network/routing/del

功能说明

删除路由

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» ids	body	string	是	路由ID	id 唯一值

Body 请求参数

```
{
  "ids": "e762817bb4e7d6889bc4cd09477672a2"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

9.8 POST /network/routing/modify

功能说明

修改路由

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» id	body	string	是	路由ID	id 唯一值
» device	body	string	是	网口名称	网口名称

名称	位置	类型	必选	中文名	说明
» ip	body	string	是	IP地址	IP地址
» netmask	body	string	是	子网掩码	子网掩码
» gw	body	string	是	网关	网关
» mtu	body	integer	是	MTU	MTU
» metric	body	integer	是	Metric	Metric

Body 请求参数

```
{
  "id": "e762817bb4e7d6889bc4cd09477672a2",
  "device": "eth1",
  "ip": "192.168.1.2",
  "netmask": "255.255.255.255",
  "gw": "192.168.36.6",
  "mtu": 1500,
  "metric": 0
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

9.9 GET /network/routing/get

功能说明

获取路由列表

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true	无	返回数据	路由列表数据
»» device	string	false	无	网口名称	网口名称
»» gw	string	false	无	网关	网关
»» id	string	false	无	路由ID	路由ID
»» ip	string	false	无	IP地址	IP地址
»» metric	integer	false	无	Metric	Metric
»» mtu	integer	false	无	MTU	MTU
»» netmask	string	false	无	子网掩码	子网掩码
»» status	integer	false	无	启用状态	状态
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok",
  "data": [
    {
      "id": 1,
      "device": "eth0",
      "ip": "192.168.38.12",
      "netmask": "255.255.255.0",
      "gw": "192.168.38.254",
      "mtu": 1500,
      "metric": 0,
      "status": 0
    }
  ]
}
```

9.10 POST /network/set

功能说明

修改网络设置

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识

名称	位置	类型	必选	中文名	说明
body	body	object	否	请求体	请求体
» dynamic	body	string	是	DHCP状态	是否DHCP; "y"=DHCP, 此时无需其他字段
» device	body	string	dynamic≠"y" 时必填	网口名称	网口名称
» ip	body	string	dynamic≠"y" 时必填	IP地址	IP地址
» mask	body	string	dynamic≠"y" 时必填	子网掩码	子网掩码
» mac	body	string	dynamic≠"y" 时必填	MAC地址	MAC地址
» gw	body	string	否	网关	网关（不校验）
» dns	body	string	否	DNS	DNS（不校验）

Body 请求参数

```
{
  "device": "eth0",
  "dynamic": "n",
  "ip": "192.168.36.6",
  "mask": "255.255.255.0",
  "gw": "192.168.36.254",
  "dns": "",
  "mac": "18:C3:F4:25:66:66"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

返回示例

```
{
  "result": "ok"
}
```

9.11 POST /network/modify

功能说明

修改维护地址

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	应用标识
body	body	object	否	请求体	请求体
» device	body	string	是	网口名称	网口名称
» enable2	body	boolean	是	维护地址启用	是否开启
» ip2	body	string	是	维护地址IP	维护地址IP
» mask2	body	string	是	维护地址子网掩码	维护地址子网掩码
» gw2	body	string	是	维护地址网关	维护地址子网网关

Body 请求参数

```
{
  "device": "eth0",
  "enable2": true,
  "ip2": "192.168.1.168",
  "mask2": "255.255.255.0",
  "gw2": ""
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	请求结果状态

10. SYSTIME模块

系统时间与时区管理模块，用于获取和设置设备系统时间、时区及NTP服务器配置。

10.1 GET /timeregion/settings

功能说明

获取时区设置信息

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true		返回数据	返回数据对象
»» ntp_servers	string	true		NTP服务器	NTP服务器地址
»» offset	integer	true		时区偏移	时区偏移
»» time	string	true		系统时间	时间字符串
»» timetype	string	true		时间类型	3种时间类型：ntp、pc、manual
»» timezone	string	true		时区名称	时区名称
» result	string	true		返回结果	操作结果

返回示例

```
{
  "data": {
    "ntp_servers": "ntp1.aliyun.com ntp2.aliyun.com ntp3.aliyun.com",
    "offset": 8,
    "time": "2024-09-09 10:40:46",
    "timetype": "pc",
    "timezone": "Asia/Shanghai"
  },
  "result": "ok"
}
```

10.2 POST /timeregion/time/set

功能说明

时间设置

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体

名称	位置	类型	必选	中文名	说明
» time	body	string	是	系统时间	时间字符串形式: "2024-09-09 09:20:46"
» timetype	body	string	是	时间类型	三种情况: ntp、pc、manual
» offset	body	integer	是	时区偏移	时区偏移, 如 8
» ntp_servers	body	string	是	NTP服务器	NTP 服务器 URL 地址, 非 ntp 时传空或不传

Body 请求参数

```
{
  "time": "2024-09-09 09:20:46",
  "timetype": "pc",
  "offset": 8,
  "ntp_servers": "ntp1.aliyun.com ntp2.aliyun.com ntp3.aliyun.com"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

10.3 POST /timeregion/timezone/set

功能说明

时区设置

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体
» timezone	body	string	是	时区名称	时区名称

名称	位置	类型	必选	中文名	说明
» offset	body	integer	是	时区偏移	时区偏移

Body 请求参数

```
{
  "timezone": "Asia/Shanghai",
  "offset": 8
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

11. SYSTEM模块

系统管理模块，提供设备重启、重置、恢复出厂设置，以及设备名称与网络配置的管理能力。

11.1 GET /sys/reboot

功能说明

系统重启

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

11.2 GET /sys/reset

功能说明

系统重置

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

11.3 GET /sys/restore

功能说明

系统恢复出厂设置

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

11.4 GET /sys/device/get

功能说明

获取设备名称

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true		返回数据	返回数据对象
»» device_name	string	true		设备名称	设备名称
» result	string	true		返回结果	操作结果

返回示例

```
{
  "data": {
    "device_name": "DECODER-D350159856666\n"
  },
  "result": "ok"
}
```

11.5 POST /sys/device/set

功能说明

设置设备名称

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌

名称	位置	类型	必选	中文名	说明
body	body	object	否	请求体	请求体
» device_name	body	string	是	设备名称	设备名称

Body 请求参数

```
{
  "device_name": "D350-211111"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

11.6 GET /sys/ip/get

功能说明

根据 URL 解析对应的 IP 和端口。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
url	query	string	是	流地址	流地址

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true		返回数据	返回数据对象
»» ip	string	true		IP地址	解析后的 IP 地址
»» path	string	true		路径	解析后的路径

名称	类型	必选	约束	中文名	说明
»» port	integer	true		端口号	解析后的端口号
» result	string	true		返回结果	操作结果

返回示例

```
{
  "data": {
    "ip": "10.20.30.21",
    "path": "",
    "port": 5961
  },
  "result": "ok"
}
```

11.7 POST /sys/ip/replace

功能说明

替换流地址中的 IP 和端口。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体
» url	body	string	是	流地址	原始流地址
» target_ip	body	string	是	目标IP	目标 IP 地址
» target_port	body	string	是	目标端口	目标端口号

Body 请求参数

```
{
  "url": "srt://192.168.1.1:13?dddddd",
  "target_ip": "192.168.1.124",
  "target_port": "1234"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true		返回数据	返回数据对象
»» url	string	true		流地址	替换后的流地址
» result	string	true		返回结果	操作结果

返回示例

```
{
  "data": {
    "url": "srt://192.168.1.124:1234?dddddd"
  },
  "result": "ok"
}
```

11.8 GET /sys/config/export

功能说明

导出设备源配置，返回 zip 文件二进制流，压缩包内为 kv_source.json（源分组与流配置）。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体对象（可选）
» modules	body	array	否	模块列表	待导出模块，当前仅支持 "source"，缺省为 ["source"]
» name	body	string	否	文件名	导出 zip 文件名（不含扩展名），缺省为 "config"

Body 请求参数（可选）

```
{
  "modules": ["source"],
  "name": "source_config"
}
```

返回数据结构

状态码 200

成功时返回 zip 文件二进制流（Content-Type: application/octet-stream），响应头携带 Content-Disposition: attachment; filename=".zip"。失败时返回 JSON：{"result": "error", "msg": "<错误码>"}。

11.9 POST /sys/config/import

功能说明

导入设备源配置。支持两种方式：通过 JSON 指定 zip 文件绝对路径（klnk-app 代理场景），或通过 multipart/form-data 直接上传 zip 文件。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体对象
» path	body	string	否	文件路径	zip 文件绝对路径（仅 /tmp、/var/tmp），与上传文件二选一
» upload	body	string(binary)	否	配置文件	multipart/form-data 上传的 zip 文件，与 path 二选一

Body 请求参数

方式一（JSON 指定路径）：

```
{
  "path": "/tmp/source_config.zip"
}
```

方式二（multipart/form-data）：文件字段名为 `upload`。

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	操作结果	请求结果，"ok" 表示成功

返回示例

```
{
  "result": "ok"
}
```

12. MAINTENANCE模块

设备维护模块，用于获取设备运行状态，包括 CPU、内存使用率以及温度等信息。

12.1 GET /maintenance/usage_get

功能说明

获取设备使用率

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌

返回数据结构

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果
» data	object	true		返回数据	返回数据对象
»» cpu_precent	number	true		CPU占用率	CPU 使用率 (%)。注意：字段名即为 cpu_precent, 非 cpu_percent。
»» mem_precent	number	true		内存占用	内存使用率 (%)
»» temperature	number	true		CPU温度	CPU 温度 (摄氏度)，-1 表示传感器不可用

状态码 200

返回示例

```
{
  "result": "ok",
  "data": {
    "cpu_precent": 0.74,
    "mem_precent": 23.02,
    "temperature": -1
  }
}
```

12.2 POST /sys/preset/set

功能说明

设置视频源无信号时的画面处理策略。当某个视频源断流或无信号输入时，设备根据此预设值决定输出画面行为。该设置全局生效，对所有视频源通道均适用。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌
» preset	body	integer	是	预设类型	0 — 无信号画面（显示预设的无信号占位图）； 1 — 黑场（输出纯黑画面）； 2 — 未预设（不做任何画面处理，等价于无操作）

Body 请求参数

```
{
  "preset": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	"ok" 表示设置成功

状态码 200（失败）

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	"error" 表示设置失败
» msg	string	true	无	错误码	401 — 缺少必要参数 preset

返回示例

```
{
  "result": "ok"
}
```

错误返回示例

```
{
  "result": "error",
  "msg": "401"
}
```

12.3 GET /sys/preset/get

功能说明

获取当前视频源无信号时的画面处理策略预设值。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	当前用户的认证令牌

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true	无	返回结果	"ok" 表示请求成功
» data	integer	true	无	预设类型	0 — 无信号画面（显示预设的无信号占位图）； 1 — 黑场（输出纯黑画面）； 2 — 未预设（不做任何画面处理）

返回示例

```
{
  "result": "ok",
  "data": 0
}
```

13. FIRMWARE模块

固件升级模块，支持通过 multipart/form-data 上传固件文件进行设备固件升级。

13.1 POST /firmware/upgrade

功能说明

固件升级

请求参数

名称	位置	类型	必选	中文名	说明
Content-Type	header	string	是	内容类型	multipart/form-data
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体

名称	位置	类型	必选	中文名	说明
» upload	body	string(binary)	是	固件文件	固件文件二进制数据
» path	body	string	否	文件路径	文件路径（可选）
» size	body	integer	否	文件大小	固件文件长度

Body 请求参数 (multipart/form-data)

upload: <固件文件二进制>
size: 文件长度

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

14. PREVIEW模块

视频预览管理模块，用于获取当前预览窗格中的视频源列表，以及向预览窗格中添加或移除视频源。

14.1 GET /preview/get

功能说明

获取预览中的视频源列表

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true		返回数据	返回数据对象
»» position	[object]	true		窗格信息	窗格位置信息数组
»»» height	integer	true		高度	窗格显示高度
»»» id	integer	true		窗格位置ID	窗格位置 ID
»»» resolution	string	true		分辨率	分辨率信息

名称	类型	必选	约束	中文名	说明
»»» status	integer	true		拉流状态	拉流状态
»»» stream_id	string	true		源ID	流 ID
»»» stream_name	string	true		流名称	流名称
»»» stream_url	string	true		流地址	流 URL
»»» type	string	true		流类型	流类型
»»» width	integer	true		宽度	窗格显示宽度
»»» x	integer	true		X坐标	图像 X 坐标位置
»»» y	integer	true		Y坐标	图像 Y 坐标位置
» result	string	true		返回结果	操作结果

返回示例

```
{
  "data": {
    "position": [
      {
        "height": 144,
        "id": 1,
        "resolution": "1920x1080p60",
        "status": 2,
        "stream_id": "e893d03a94326b981fd3aa276f90183a",
        "stream_name": "N60_238 (HB)",
        "stream_url": "192.168.36.238:5965",
        "type": "ndi",
        "width": 320,
        "x": 0,
        "y": 0
      },
      {
        "height": 144,
        "id": 2,
        "resolution": "640x360p30",
        "status": 2,
        "stream_id": "0f79952dbc00505b2331b0bdb0c8ca0a",
        "stream_name": "N60_238 (HX)",
        "stream_url": "192.168.36.238:5964",
        "type": "ndi",
        "width": 320,
        "x": 320,
        "y": 0
      }
    ]
  },
  "result": "ok"
```

```
}
```

14.2 POST /preview/source/modify

功能说明

添加视频源到预览中

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体
» from	body	object	是	源信息	来源信息
»» pos_id	body	string	是	窗格位置ID	来源窗格位置 ID（实际生效字段）
»» type	body	string	否	来源类型	来源类型
»» stream_id	body	string	否	源ID	流 ID
»» stream_name	body	string	否	流名称	流名称
»» stream_url	body	string	否	流地址	流 URL
»» output_id	body	string	否	输出ID	输出 ID
»» layout_id	body	string	否	布局ID	布局 ID
» to	body	object	是	目标信息	目标信息
»» stream_id	body	string	是	源ID	目标流 ID（实际生效字段）
»» pos_id	body	string	是	窗格位置ID	目标窗格位置 ID（实际生效字段）
»» type	body	string	否	目标类型	加入类型
»» stream_name	body	string	否	流名称	流名称
»» stream_url	body	string	否	流地址	流 URL
»» output_id	body	string	否	输出ID	输出 ID
»» layout_id	body	string	否	布局ID	布局 ID

说明：本接口实际只使用 `from.pos_id`、`to.stream_id`、`to.pos_id` 三个字段，其余字段可不传。

Body 请求参数

```
{
  "from": {
    "type": "source",
```

```
    "stream_id": "0f79952dbc00505b2331b0bdb0c8ca0a",
    "stream_name": "N60_238 (HX)",
    "stream_url": "192.168.36.238:5964",
    "pos_id": "",
    "output_id": "1",
    "layout_id": ""
  },
  "to": {
    "type": "preview",
    "stream_id": "0f79952dbc00505b2331b0bdb0c8ca0a",
    "stream_name": "N60_238 (HX)",
    "stream_url": "192.168.36.238:5964",
    "output_id": "1",
    "layout_id": ""
  }
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

14.3 POST /preview/source/remove

功能说明

删除预览中的视频源

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体
» pos_id	body	integer	是	窗格位置ID	窗格位置 ID（字段名为 pos_id，非 id）

Body 请求参数

```
{
  "pos_id": 1
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

15. NDI-Discovery模块

NDI 发现服务管理模块，管理 NDI 源的自动发现、手动发现、按组自动发现和服务器发现四种发现方式及其对应配置。

15.1 GET /ndi/discovery/all

功能说明

获取所有发现类型及配置

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	object	true		返回数据	返回数据对象
»» auto_server	[object]	true		自动/服务器发现	自动发现与服务器发现列表（含 dis_method=1/3/4 的项）
»»» dis_d_ip	string	true		手动发现IP	手动发现目标 IP
»»»» dis_g_name	string	true		组名称	组名称
»»»» dis_method	integer	true		发现方式	1=自动发现，2=手动发现，3=按组自动发现，4=服务器发现
»»»» dis_s_ip	string	true		发现服务器 IP	发现服务器 IP

名称	类型	必选	约束	中文名	说明
»»» id	string	true		配置项ID	配置项唯一 ID
»» manual	[object]	true		手动发现	手动发现列表（仅 dis_method=2 的项）
»»» dis_d_ip	string	true		手动发现IP	手动发现目标 IP
»»» dis_g_name	string	true		组名称	组名称
»»» dis_method	integer	true		发现方式	发现方式，参见 auto_server 同名字段说明
»»» dis_s_ip	string	true		发现服务器 IP	发现服务器 IP
»»» id	string	true		配置项ID	配置项唯一 ID
» result	string	true		返回结果	操作结果

注意：data 下只有 manual 与 auto_server 两个数组（dis_method=1 自动发现、dis_method=3 按组自动发现、dis_method=4 服务器发现都归入 auto_server），且每一项都含 id。不存在 auto/discovery_server/public 这些键。

返回示例

```
{
  "data": {
    "manual": [
      {
        "dis_d_ip": "192.168.43.141",
        "dis_g_name": "public",
        "dis_method": 2,
        "dis_s_ip": "",
        "id": "1"
      },
      {
        "dis_d_ip": "192.168.35.178",
        "dis_g_name": "mhb",
        "dis_method": 2,
        "dis_s_ip": "",
        "id": "2"
      }
    ],
    "auto_server": [
      {
        "dis_d_ip": "",
        "dis_g_name": "123",
        "dis_method": 3,
        "dis_s_ip": "",
        "id": "3"
      }
    ]
  }
}
```

```
    },
    {
      "dis_d_ip": "",
      "dis_g_name": "public",
      "dis_method": 4,
      "dis_s_ip": "192.168.43.141",
      "id": "4"
    }
  ]
},
"result": "ok"
}
```

15.2 POST /ndi/discovery/auto/del

功能说明

删除自动发现项

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体
» id	body	string	是	配置项 ID	唯一 ID（取自 /ndi/discovery/all 返回项的 id，为字符串）

Body 请求参数

```
{
  "id": "1"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

15.3 GET /ndi/discovery/sources

功能说明

获取对应发现方式发现的 NDI 源列表

请求参数

名称	位置	类型	必选	中文名	说明
dis_method	query	integer	是	发现方式	1=自动发现, 2=手动发现, 3=按组自动发现, 4=服务器发现 (必填, 缺失返回错误码 1)
dis_g_name	query	string	否	组名称	组名称, dis_method=3、4 时需要此参数
dis_s_ip	query	string	否	发现服务器IP	发现服务器 IP, dis_method=4 时需要此参数
group_id	query	string	否	分组ID	分组 ID, 用于按分组筛选源列表
app	header	string	是	鉴权令牌	鉴权令牌

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» data	[object]	true		返回数据	返回数据数组
»» channel	string	false		通道名称	通道名称
»» group_name	string	false		组名	组名
»» ip	string	false		IP地址	IP 地址
»» name	string	false		设备名	设备名称
»» ndi_name	string	false		NDI全名	NDI 发现的完整名称
»» port	integer	false		端口号	端口号
»» url	string	false		流地址	源 URL
» result	string	true		返回结果	操作结果

返回示例

```
{
  "data": [
    {
      "channel": "FULL",
```

```
    "group_name": "public",
    "ip": "192.168.35.178",
    "name": "N60-N6020240531 ",
    "ndi_name": "N60-N6020240531 (FULL)",
    "port": 5961,
    "url": "192.168.35.178:5961"
  }
],
"result": "ok"
}
```

15.4 POST /ndi/discovery/manual/add

功能说明

添加或更新手动发现配置项

说明：此接口也作为删除配置项接口使用——用户在页面删除相应 IP/组后，将剩余的配置项整体传给后台即可，与矩阵操作方式一致。

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体
» manuals	body	[object]	是	手动配置列表	手动配置项数组
»» dis_d_ip	body	string	是	发现IP	目标发现 IP
»» dis_g_name	body	string	是	组名称	发现组名称

Body 请求参数

```
{
  "manuals": [
    {
      "dis_d_ip": "192.168.43.141",
      "dis_g_name": "public"
    }
  ]
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

15.5 POST /ndi/discovery/auto/add

功能说明

添加自动发现项（dis_method=3，按组自动发现）

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体
» dis_g_name	body	string	是	组名称	组名称

Body 请求参数

```
{
  "dis_g_name": "123"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

15.6 POST /ndi/discovery/server/del

功能说明

删除服务器发现项

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体
» id	body	string	是	配置项 ID	配置项唯一 ID（取自 /ndi/discovery/all 返回项的 id，为字符串）

Body 请求参数

```
{
  "id": "1"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```

15.7 POST /ndi/discovery/server/add

功能说明

添加服务器发现项（dis_method=4，通过指定 NDI 发现服务器来发现源）

请求参数

名称	位置	类型	必选	中文名	说明
app	header	string	是	鉴权令牌	鉴权令牌
body	body	object	否	请求体	请求体
» dis_g_name	body	string	是	组名称	组名称
» dis_s_ip	body	string	是	服务器IP	NDI 发现服务器 IP

Body 请求参数

```
{
  "dis_g_name": "public",
  "dis_s_ip": "192.168.43.141"
}
```

返回数据结构

状态码 200

名称	类型	必选	约束	中文名	说明
» result	string	true		返回结果	操作结果

返回示例

```
{
  "result": "ok"
}
```