

API 说明文档

N60 视频编解码器

NDI High Bandwidth+NDI|HX

全能编解码器

(V2.0 版)



通过本手册，您可以了解到

- API 的接口说明
- API 的使用方法

由于产品不断升级更新，您所购买的产品可能与本手册的内容有所变更，请以包装箱内产品实际为准。

法律声明

若接收长沙千视电子科技有限公司（以下称为“千视电子”）的此份文档，即表示您已同意以下条款。若不同意以下条款，请停止使用本文档。

本文档版权所有长沙千视电子科技有限公司。保留任何未在本文档中明示授予的权利。文档中涉及千视电子的专有信息。未经千视电子事先书面许可，任何单位和个人不得复制、传递、分发、使用和泄漏该文档以及该文档包含的任何图片、表格、数据及其他信息。

 是千视电子的注册商标。千视电子产品的名称和标志是千视电子的商标或注册商标。在本文档中提及的其他产品或公司名称可能是其各自所有者的商标或注册商标。在未经千视电子或第三方权利人事先书面同意的情况下，阅读本文档并不表示以默示、不可反言或其他方式授予阅读者任何使用本文档中出现的任何标记的权利。

本产品符合有关环境保护和人身安全方面的设计要求，产品的存放、使用和弃置应遵照产品手册、相关合同或相关国法律、法规的要求进行。

本文档按“现状”和“仅此状态”提供。本文档中的信息随着千视电子产品和技术的进步将不断更新，千视电子不再通知此类信息的更新。

本文档未尽事宜，请访问千视电子网站 www.kiloview.com 获取相关信息和技术支持。

文档修订说明：

版本号	修订日期	修订说明	修订人	审批人
V2.0	2023-12-14	版本升级，基于 2.02.0008 版 N60 固件；新增支持录像，支持 SRT 协议		

目录

1 系统模块.....	6
1.1 用户管理.....	6
1.1.1 用户登录.....	6
1.1.2 登录校验.....	7
1.1.3 获取所有用户.....	7
1.1.4 添加用户.....	8
1.1.5 删除用户（可以单个删除或者批量删除）.....	9
1.1.6 修改用户信息.....	9
1.2 系统时区.....	10
1.2.1 设置地区时区.....	10
1.2.2 设置系统时间.....	10
1.3 系统配置.....	11
1.3.1 获取开源协议.....	11
1.3.2 获取设备名称.....	12
1.3.3 设置设备名称.....	12
1.3.4 获取 http 和 https 端口开关.....	13
1.3.5 设置 http 和 https 端口开关.....	13
1.3.6 恢复出厂设置.....	14
1.3.7 复位.....	14
1.3.8 重启设备.....	15
1.3.9 获取网卡速度.....	15
1.3.10 获取系统信息.....	16
1.3.11 设置序列号.....	17
1.3.12 上传升级固件.....	17
1.3.13 测试页面服务的开关（N60）.....	18
1.3.14 测试页面服务的启停状态（N60）.....	18
2 视频编解码.....	19
2.1 解码.....	19
2.1.1 选择当前解码流.....	19
2.1.2 从预设中选择解码流.....	20
2.1.3 获取解码信息.....	20
2.1.4 设置输出的解码信息.....	24
2.1.5 设置 NDI 连接方式.....	26
2.2 发现服务模块.....	26
2.2.1 设置发现服务器.....	26
2.2.2 获取发现服务器.....	27
2.2.3 扫描获取 NDI Source 列表.....	27
2.2.4 手动添加源配置.....	29
2.2.5 获取手动添加的 NDI 源地址.....	29
2.2.6 设置解码模式下的发现参数.....	30
2.2.7 获取该设备在解码模式下的发现参数.....	30

2.3	预设模块	31
2.3.1	获取预设列表	31
2.3.2	添加预设	32
2.3.3	删除预设	33
2.3.4	设置背景颜色	33
2.4	解码 PTZ 相机控制	34
2.4.1	解码模式 PTZ 控制	34
2.5	编码	35
2.5.1	获取所有编码对象名称	35
2.5.2	选择输出的编码模式	35
2.5.3	获取码流编码参数	36
2.5.4	获取视频源参数	37
2.5.5	设置主码流 (NDI HX) 编码参数	38
2.5.6	设置 FULL NDI 流 (NDI HB) 编码参数	39
2.5.7	设置多协议流编码参数	40
2.5.8	获取流信息	41
2.5.9	所有流重新推流	42
2.5.10	获取流相关属性及配置	42
2.5.11	获取流实时码率、连接数	44
2.5.12	更新流相关配置	45
2.6	模式	46
2.6.1	设置工作模式 (编码模式 解码模式)	46
2.6.2	获取工作模式	46
2.7	编码 PTZ 控制	47
2.7.1	编码 PTZ 控制	47
2.7.2	获取 PTZ 信息	47
2.7.3	获取 PTZ 协议	49
2.7.4	获取串口设备	49
2.7.5	修改 PTZ 信息	50
2.8	二维码扫描	51
2.8.1	虚拟用户身份登录	51
2.8.2	获取二维码的 ID	51
2.8.3	二维码校验	52
2.8.4	检查当前是否为解码模式	52
2.8.5	获取二维码	53
2.9	页面预览画面 (WebRTC)	54
2.9.1	获取 webrtc 的 sdp 和 port	54
2.9.2	设置 webrtc 的模式	54
3	网络	55
3.1	网络管理	55
3.1.1	获取全部网络信息	55
3.1.2	修改当前设备网络信息	56
3.1.3	以太网开关	57
3.1.4	获取网卡信息	57
3.1.5	设置网卡信息	58
3.1.6	设置链路聚合	59
3.1.7	获取链路聚合信息	60
3.1.8	重启 MDNS	61
3.2	静态路由管理	61
3.2.1	添加静态路由	61
3.2.2	获取静态路由列表	62

3.2.3	修改静态路由	63
3.2.4	删除静态路由	64
3.2.5	静态路由开关	64
4	Kilolink	65
4.1	Kilolink 客户端服务	65
4.1.1	获取当前设备网络接口	65
4.1.2	获取 kilolink client 设置	65
4.1.3	设置 kilolink client 配置参数	66
4.1.4	校验 token	67
5	语音对讲	67
5.1	Talk server	67
5.1.1	添加设备	67
5.1.2	设置本机设备与目标设备的连接	68
5.1.3	设置网页与目标设备的连接	68
5.1.4	设置设备和语音服务器的连接	69
5.1.5	查询设备连接状态	69
5.1.6	查询设备列表	70
5.1.7	删除设备	71
5.1.8	查询本机设备的序列号	71
5.1.9	talk server 版本	72
5.1.10	设置语言	72
6	音视频录制	73
6.1	存储管理	73
6.1.1	添加 nas 网盘	73
6.1.2	移除 nas 网盘	74
6.1.3	获取已挂载磁盘列表	74
6.1.4	获取录制的文件列表	76
6.1.5	删除录制文件	77
6.2	录制配置管理	78
6.2.1	录制信息设置	78
6.2.2	获取录制信息	78
6.3	录制	79
6.3.1	录制启停	79

1 系统模块

1.1 用户管理

1.1.1 用户登录

- 接口描述

单独请求该接口并不能直接登录，需要拿着接口下返回的 token 和 alias 值，结合 username 构造 cookie，并将其加入到请求头中请求 1.1.2 的校验接口，校验通过后才能登录。

- 请求地址

/api/systemctrl/users/login

- 请求方法

POST

- 请求参数（样例）

```
{
  "username": "admin",
  "password": "Admin123"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
App	{"language": "en"}	Y	[STRING]	请求头参数，目前版本只支持中英文语言，分别为 "en"/"zh"，用于弹窗提示的多语言输出。 注意： 当前版本中，如果单独请求此接口时想要返回正确的错误提示，请务必在请求头中加入此参数。否则返回的提示中 msg 字段将为 null
username	"admin"	Y	[STRING]	用户名
password	"Admin123"	Y	[STRING]	用户密码

- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "token": "0621da14f9d6546688cf87c5eab9c2f9",
    "alias": "Admin",
    "changed": true
  },
  "msg": null
}
```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
token	[STRING]	登录签发的 token，用于登录保持
alias	[STRING]	别名
changed	[BOOLEAN]	用户是否已修改初始密码。未修改初始密码，前端跳转修改初始密码页面。

1.1.2 登录校验

- 接口描述

用于正常登录请求，和 kilolink 反向代理过来请求两方面的 session 校验和验证放行。正常登录请求需要在请求头中添加 cookie 并组装字段值，参考样例；kilolink 反向代理的则需要在请求中添加 platform 字段，并指定相应的字段标识符，参考样例。

- 请求地址

/api/systemctrl/users/session

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
platform	"this-is-made-by-kiloview-for-platform-login"	Y	[STRING]	请求参数，如果想单独使用此接口，需要用户将该值单独加入到 kilolink server 端请求头中。
Cookie	"language=zh; user=kiloview; alias=Kiloview; token=18532c790a58dcc31b0d473b3a791770"	Y	[STRING]	请求头参数，如果想单独使用此接口，需要用户将该值单独加入到请求头中。

- 响应参数（样例）

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

- 响应参数说明

无

1.1.3 获取所有用户

- 接口描述

请求该接口可以获取到当前设备上保留的所有可登录用户

- 请求地址

/api/systemctrl/users/list

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

```
{
  "result": "ok",
  "data": [
    {
      "id": "admin",
      "alias": "Admin",
      "web": true,
      "api": true,
      "create_time": "1970-01-01 00:00:00"
    }
  ]
}
```

- 响应参数说明（data 字段中的单个元素各字段说明）

字段名	字段类型	字段描述
id	[STRING]	用户名
alias	[STRING]	别名
web	[BOOLEAN]	是否允许在 web 端新增用户
api	[BOOLEAN]	-
create_time	[STRING]	创建时间

1.1.4 添加用户

- 接口描述
由管理员或者其他已经登录的用户添加新用户的接口
- 请求地址
/api/systemctrl/users/add
- 请求方法
POST
- 请求参数（样例）

```
{
  "id": "test_user",
  "password": "qwer1234",
  "confirmNewPassword": "qwer1234",
  "alias": "test_user",
  "enable_web": true,
  "enable_api": true
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
id	"test_user"	Y	[STRING]	用户名
alias	"test_user"	Y	[STRING]	别名
password	"qwer1234"	Y	[STRING]	用户密码
confirmNewPassword	"qwer1234"	Y	[STRING]	确认密码
enable_web	true	Y	[BOOLEAN]	是否允许在 web 端新增用户
enable_api	true	Y	[BOOLEAN]	-

- 响应参数（样例）

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

- 响应参数说明
无

1.1.5 删除用户（可以单个删除或者批量删除）

- 接口描述
在请求该接口时，传入单个用户名，或者用户名组成的列表，随后即可删除相应的用户。

- 请求地址
/api/systemctrl/users/remove

- 请求方法
POST

- 请求参数
单个删除样例

```
{
  "ids": "Jack"
}
```

批量删除样例

```
{
  "ids": ["Jack", "Bob"]
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
ids	"Jack" / ["Jack", "Bob"]	Y	[STRING]/[ARRAY(STRING)]	用户名 或 用户名列表

- 响应参数（样例）

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

- 响应参数说明
无

1.1.6 修改用户信息

- 接口描述
请求此接口，可以修改设备上已经保存的之外的用户信息，但是只能修改昵称。

- 请求地址
/api/systemctrl/users/modify

- 请求方法
POST

- 请求参数（样例）

```
{
  "id": "test1",
  "alias": "test11",
  "enable_web": true,
  "enable_api": true
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
id	"test1"	Y	[STRING]	用户名
alias	"test11"	Y	[STRING]	别名
enable_web	true	Y	[BOOLEAN]	是否允许在 web 端 新增用户

enable_api	true	Y	[BOOLEAN]	-
------------	------	---	-----------	---

- 响应参数（样例）

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

- 响应参数说明
无

1.2 系统时区

1.2.1 设置地区时区

- 接口描述

请求此接口用于设置地区时区，需要传入指定地区的标准时字符串

- 请求地址

/api/systemctrl/systime/setLocation

- 请求方法

POST

- 请求参数（样例）

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

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
timezone	"Asia/Shanghai"	Y	[STRING]	时区
offset	8	N	[INT]	当 timezone 无法找到时,会使用此字段,此字段为格林威治时间基础上加减的小时,如 GMT+8, 范围为-14 到 12

- 响应参数（样例）

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

- 响应参数说明
无

1.2.2 设置系统时间

- 接口描述

请求此接口设置系统时间。可以选择与当前 PC 对时、手动校时，或者从 NTP 服务器同步三种方式设置。

- 请求地址

/api/systemctrl/systime/setTime

- 请求方法

POST

- 请求参数（样例）

无

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
timetype	"pc"	Y	[STRING]	时间设置方式 pc: 与当前 PC 对时; manual: 手动校时; ntp: 从 NTP 服务器同步
time	"2023-08-07 17:37:14"	N	[STRING]	标准格式的时间, 格式为 "Y-m-d H:M:S"。 当 timetype 参数值为 pc 或 manual 时, 必填!
ntp	"ntp1.aliyun.com ntp2.aliyun.com ntp3.aliyun.com"	N	[STRING]	ntp 服务器地址, 多个服务器之前用空格间隔。 当 timetype 参数值为 ntp 时, 必填!

- 响应参数 (样例)

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

- 响应参数说明
无

1.3 系统配置

1.3.1 获取开源协议

- 接口描述

请求此接口或许开源协议, 固定值。

- 请求地址
`/api/systemctrl/system/components`
- 请求方法
GET
- 请求参数 (样例)
无
- 请求参数说明
无
- 响应参数

```
{
  {
    "name": "vue",
    "url": "MIT",
    "software": "MIT License",
    "address": "https://github.com/vuejs/vue"
  }, {
    "name": "Element-UI",
    "url": "MIT",
    "software": "MIT License",
    "address": "https://github.com/ElementFE/element"
  },
  {
    "name": "python3",
    "url": "PSF",
    "software": "PSF",
    "address": "http://www.python.org"
  },
  {

```

```

        "name": "nginx",
        "url": "BSD2C",
        "software": "BSD-2c",
        "address": "http://nginx.org"
    }, {
        "name": "libev",
        "url": "BSD2C",
        "url2": "GPL",
        "software": "BSD-2c",
        "software2": "GPLv2+",
        "address": "http://dist.schmorp.de/libev"

    }, {
        "name": "openssl",
        "url": "OpenSSL",
        "url2": "SSLeay",
        "software": "OpenSSL",
        "software2": "SSLeay",
        "address": "http://www.openssl.org"
    }
}
    
```

- 响应参数说明
无

1.3.2 获取设备名称

- 接口描述
请求此接口获取设备的名称，该名称通常由用户指定，没有手动设置的情况下，默认为设备型号+序列号。
- 请求地址
/api/systemctrl/system/getHostname
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```

{"result":"ok", "data":{"hostname": "N60-123456"}}
    
```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
hostname	[STRING]	当前产品的名称

1.3.3 设置设备名称

- 接口描述
通过此接口可以设置设备的名称。
- 请求地址
/api/systemctrl/system/setHostname
- 请求方法
GET
- 请求参数（样例）

无

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
name	"N60"	Y	[STRING]	设备名称

- 响应参数（样例）

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

- 响应参数说明

无

1.3.4 获取 http 和 https 端口开关

- 接口描述

此接口提供给前端，用户获取当前设备上保存的 http 和 https 的端口开关开闭情况，并显示在页面上。

- 请求地址

/api/systemctrl/system/getProtocol

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

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

- 响应参数说明

字段名	字段类型	字段描述
http	[BOOLEAN]	是否开启 http 连接
https	[BOOLEAN]	是否开启 https 连接
httpPort	[INT]	http 端口号
httpsPort	[INT]	https 端口号

1.3.5 设置 http 和 https 端口开关

- 接口描述

默认情况下，本机器 https 不开启，如果想要使用语音对讲功能，可以通过此接口打开 https 的开关

- 请求地址

/api/systemctrl/system/setProtocol

- 请求方法

POST

- 请求参数（样例）

```
{
"http": true,
"https": true,
"httpPort": 80,
"httpsPort": 443
}
// http 和 https 不能同时关闭
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
http	true	Y	[BOOLEAN]	是否开启 http 连接
https	true	Y	[BOOLEAN]	是否开启 https 连接
httpPort	80	Y	[INT]	设置 http 端口号
httpsPort	443	Y	[INT]	设置 https 端口号

- 响应参数（样例）

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

- 响应参数说明
无

1.3.6 恢复出厂设置

- 接口描述

通过请求该接口，后期所有在当前设备上做的设置将被清除，恢复到当前版本的全新状态。

- 请求地址
/api/systemctrl/system/reFactory
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

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

- 响应参数说明
无

1.3.7 复位

- 接口描述

通过请求该接口，可以重启 codec 编解码服务。

- 请求地址
/api/systemctrl/system/restore
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无

- 响应参数（样例）

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

- 响应参数说明
无

1.3.8 重启设备

- 接口描述

通过请求该接口，设备将立即重启

- 请求地址

/api/systemctrl/system/reboot

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

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

- 响应参数说明
无

1.3.9 获取网卡速度

- 接口描述

当前设备上有 eth0 和 eth1 两张网卡。在请求时该接口时，可以选择指定网卡返回速度。如果不给参数，则默认将两张网卡的速度都返回给用户。

- 请求地址

/api/systemctrl/system/getNetworkSpeed

example:

/api/systemctrl/system/getNetworkSpeed?ifname=eth0

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

参数名称	参数值	是否必填	参数类型	参数描述
ifname	"eth0"/"eth1"	N	[STRING]	当请求给了该参数时，将按照当前参数返回对应的网卡速度。 未填写时，eth0 和 eth1 都返回

- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "eth0": {
      "receive": 18.55,
```

```

        "transmit": 433.52,
        "speed": 1000
    },
    "eth1": {}
}
    
```

- 响应参数说明（data 字段中每张网卡的说明）

字段名	字段类型	字段描述
receive	[NUMBER]	两次接口调用间的接收数据量
transmit	[NUMBER]	发送数据量
speed	[NUMBER]	网卡速度

1.3.10 获取系统信息

- 接口描述
通过给定的参数来获取指定的信息，当前版本设备参数可在设备固件版本、cpu 使用率、内存的使用率、磁盘使用率、持续时间中进行选择开启或者关闭

- 请求地址
/api/systemctrl/system/getSystemInfo
example:
/api/systemctrl/system/getSystemInfo?version=true&cpu=true&memory=true&disk=true&persisTime=true

- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
version	true	N	[BOOLEAN]	当前设备的固件版本
cpu	true	N	[BOOLEAN]	获取 cpu 的使用率
memory	true	N	[BOOLEAN]	获取 memory 的使用率
disk	true	N	[BOOLEAN]	获取 disk 的使用率
persisTime	true	N	[BOOLEAN]	获取 persisTime 的使用率

- 响应参数（样例）

```

{
    "result": "ok",
    "data": {
        "version": {
            "ndiVersion": "5.1.3",
            "hardwareVersion": "1.0",
            "softwareVersion": "1.0.1234",
            "serialNumber": "00000000",
            "product": "",
            "oem": ""
        },
        "cpu": {
            "precent": 2.624
        },
        "mem": {
            "precent": 16.282,
            "total": 4039368,
            "used": 248088
        }
    }
}
    
```

```

    },
    "disk": {
      "precent": 28,
      "free": 1350944,
      "used": 513328,
      "total": 1983056
    },
    "persisTime": {
      "uptime": 1933.73
    }
  }
}

```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
version	[OBJECT]	版本信息,软件版本,硬件版本,NDI 版本,序列号
cpu	[OBJECT]	cpu 利用率(%)
memory	[OBJECT]	内存使用率 precent(%),内存总量 total(kb),内存使用量 used(kb)
disk	[OBJECT]	硬盘利用率 precent(%),硬盘总量 total(kb),硬盘剩余 free(kb),硬盘使用量 used(kb)
persisTime	[OBJECT]	系统运行时间 uptime(s)

1.3.11 设置序列号

- 接口描述

设备出厂时会指定序列号，可以通过该接口对设备序列号进行修改。

- 请求地址

/api/systemctrl/system/Rpj1Tn92uH3UiCCLPQeyLMd73hElBj6G/setSerial

example:

/api/systemctrl/system/Rpj1Tn92uH3UiCCLPQeyLMd73hElBj6G/setSerial?serial_number=7116110002301

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
serial_number	"7116110002301"	Y	[STRING]	设备序列号

- 响应参数（样例）

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

- 响应参数说明

无

1.3.12 上传升级固件

- 接口描述

该接口用于上传升级固件，该接口在单独使用时需要构造文件参数才可以继续。

- 请求地址
/api/systemctrl/system/upload
- 请求方法
POST
- 请求参数（样例）
无
- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
file	-	Y	multipart/form-data	bin 文件

- 响应参数（样例）

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

- 响应参数说明
无

1.3.13 测试页面服务的开关（N60）

- 接口描述
该接口用于提供给 webui-production 页面，开启 py-production-module 的开关

- 请求地址
/api/systemctrl/system/production_off
- 请求方法
POST
- 请求参数（样例）

```
{"value": true}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
value	true	Y	[BOOLEAN]	产品测试服务的开关。此处标识设置开关

- 响应参数（样例）

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

- 响应参数说明（data 字段说明）

参数名称	参数值样例	是否必填	参数类型	参数描述
value	true	Y	[BOOLEAN]	产品测试服务的开关。此处反映开关是否激活

1.3.14 测试页面服务的启停状态（N60）

- 接口描述
用于获取 production 服务的开关状态
- 请求地址
/api/systemctrl/system/production_stat
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无

- 响应参数（样例）

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

- 响应参数说明（data 字段说明）

参数名称	参数值样例	是否必填	参数类型	参数描述
value	true	Y	[BOOLEAN]	产品测试服务模块的状态。返回当前服务是否有激活。

2 视频编解码

2.1 解码

2.1.1 选择当前解码流

- 接口描述

解码模式下选择解码流有两种方式，第一种是直接选择发现列表中的当前解码流，第二种是选择预设中的解码流。此处 API 功能为第一种。

- 请求地址

/api/codec/decode/addSpec

- 请求方法

POST

- 请求参数（样例）

```
{
  "id": 2,
  "name": "N60 (Decoding Channel)",
  "url": "192.168.42.249:5961",
  "group": "public"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
id	2	N	[INT]	此参数是用来传递当前 NDI Source 列表中当前 NDI Source 所处的序号，暂时不强制要求该值，作为备用
name	"N60 (Decoding Channel)"	Y	[STRING]	NDI Source Name，NDI Source 的原始名称
url	"192.168.42.249:5961"	Y	[STRING]	NDI Source URL。请注意：目前 URL 的格式为 <IP>:<port>，但是建议您不要认为它就是事实，NDI SDK 并不担保这一点。在使用时，请保持这种 URL 格式，不要尝试改变它。
group	"public"	Y	[STRING]	NDI Source 所在的 NDI Group。如果您传递的 Group 是空字符串""，按照 NDI 的规则，它表示默认 public group。

- 响应参数（样例）

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

- 响应参数说明
无

2.1.2 从预设中选择解码流

- 接口描述

解码模式下选择解码流有两种方式，第一种是直接选择发现列表中的当前解码流，第二种是选择预设中的解码流。此处 API 功能为第二种。

- 请求地址
/api/codec/decode/add
- 请求方法
POST
- 请求参数（样例）

```
{"id":2}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
id	2	Y	[INT]	指定 Preset ID (0~9), 将 Preset ID 所保存的 NDI Source 切换到当前解码输出。 如果您指定 id = "0", 表示当前不解码而直接输出 Blank。

- 响应参数（样例）

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

- 响应参数说明
无

2.1.3 获取解码信息

- 接口描述

该接口通过 get 请求返回了基于当前解码源条件下解码页面上显示的一切信息，包括当前解码源的源分辨率、帧率、声道数，以及解码输出所设置声道数，输出声道和源的映射等参数值。

- 请求地址
/api/codec/decode/get
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "name": "DESKTOP-R6TEOQE (vMix - Output 1)",
    "ip": "192.168.38.238",
    "online": true,
    "resolution": "1920x1080p 60Hz",
```

```
"codec": "SHQ2",
"bitrate": 132056,
"deInterlace": false,
"interlaced": 0,
"frame_rate": 60,
"audio": "48.0KHz / 2CH",
"url": "192.168.38.238:5961",
"xRes": 1920,
"yRes": 1080,
"audio_sampling": 48000,
"audio_channels": 2,
"hdcp": 1,
"out_colorspace": 0,
"supported": true,
"output_resolution_choose": "1920x1080",
"output_resolution": "1920x1080p 60Hz",
"hdmi_channels": 2,
"hdmi_channels_mapping": [
  {
    "output_channel": 1,
    "source_channel": 1
  },
  {
    "output_channel": 2,
    "source_channel": 2
  },
  {
    "output_channel": 3,
    "source_channel": 3
  },
  {
    "output_channel": 4,
    "source_channel": 4
  },
  {
    "output_channel": 5,
    "source_channel": 5
  },
  {
    "output_channel": 6,
    "source_channel": 6
  },
  {
    "output_channel": 7,
    "source_channel": 7
  },
  {
    "output_channel": 8,
    "source_channel": 8
  }
],
"line_out_channels": 2,
"line_out_channels_mapping": [
  {
```

```

        "output_channel": 1,
        "source_channel": 1
    },
    {
        "output_channel": 2,
        "source_channel": 2
    },
    {
        "output_channel": 3,
        "source_channel": 3
    },
    {
        "output_channel": 4,
        "source_channel": 4
    },
    {
        "output_channel": 5,
        "source_channel": 5
    },
    {
        "output_channel": 6,
        "source_channel": 6
    },
    {
        "output_channel": 7,
        "source_channel": 7
    },
    {
        "output_channel": 8,
        "source_channel": 8
    }
],
"ndi_connection": "tcp"
}
}

```

● 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
name	[STRING]	当前正在解码的 NDI Source Name。如果当前正在输出 Preset 0 (Blank)，则 name = "(Blank)"
ip	[STRING]	当前正在解码的 NDI Source 的 IP 地址。如果没有解码，则 ip = "0.0.0.0"
online	[BOOLEAN]	true: 当前 NDI Source 在线； false: 当前 NDI Source 不在线
resolution	[STRING]	当前 NDI Source 的视频分辨率（友好名称）。如果您需要获取视频的程序友好的信息，例如宽度/高度等，请参考 `xRes`、`yRes`、`frame_rate`、`interlaced` 等字段。
codec	[STRING]	当前 NDI Source 的编码格式。如果源是离线状态，则会显示 "UNKNOWN"

bitrate	[INT]	当前实时统计的 bitrate, Kbps
deInterlace	[BOOLEAN]	当前 NDI Source 是否有去交织。注意: 在现阶段版本中, 该字段默认为不去交织。
interlaced	[INT]	当前 NDI Source 是否的交织模式。如果是 p, 则该值为 0; 是 i, 则该值为 1.
frame_rate	[INT]	当前 NDI Source 的原始帧率。
audio	[STRING]	当前 NDI Source 的原始音频采样率和声道数。
url	[STRING]	当前正在解码的 NDI Source 的原始 URL。如果没有解码 NDI Source, 则 url = "" (空字符串)
xRes	[INT]	当前 NDI Source 的视频分辨率宽度
yRes	[INT]	当前 NDI Source 的视频分辨率高度
audio_sampling	[INT]	当前解码后音频输出的采样率。 (此字段在当前版本并无实际作用, 后续版本将用于返回输出的采样率)
audio_channels	[INT]	当前解码后音频输出的采样率。 (此字段被下列的 hdmi 和 line-out 声道数取代, 预备废弃)
hdcp	[INT]	是否开启高清数字内容保护。1 为开启, 0 为关闭。
out_colorspace	[INT]	当前解码后输出的色彩空间类型。 0: RGB444 1: YUV422 2: YUV444
supported	[BOOLEAN]	当前 NDI Source 是否被设备支持解码。
output_resolution_choose	[STRING]	当前系统内设置的解码视频分辨率。
output_resolution	[STRING]	当前解码后实际输出的视频分辨率。
hdmi_channels	[INT]	当前解码后 hdmi 音频声道数
hdmi_channels_mapping	[OBJECT]	当前解码后 hdmi 音频声道和 NDI Source 音频声道编号的关系映射。注意声道数用阿拉伯数字标识。基本形式为: [{"output_channel": 1, "source_channel": 1}] "output_channel" 为输出声道编号; "source_channel" 为 NDI Source 音频声道编号。
line_out_channels	[INT]	当前解码后 line-out 音频声道数
line_out_channels_mapping	[OBJECT]	当前解码后 line-out 音频声道和 NDI Source 音频声道编号的关系映射。注意声道数用阿拉伯数字标识。基本形式为:

		<pre> [{"output_channel": 1,"source_channel": 1}] "output_channel"为输出声道编号; "source_channel" 为 NDI Source 音 频声道编号。 </pre>
ndi_connection	[STRING]	当前 NDI Source 的连接方式。

2.1.4 设置输出的解码信息

- 接口描述

该接口专门设置输出的解码信息，目前版本包括设置其中的分辨率，hdmi 声道数以及映射关系，line-out 声道数和映射关系。

- 请求地址

/api/codec/decode/output_set

- 请求方法

POST

- 请求参数（样例）

```

{
"output_resolution": "1920x1080",
"hdmi_channels": 2,
"output_framerate": 0,
"hdmi_channels_mapping": [{
"output_channel": 1,
"source_channel": 1
}, {
"output_channel": 2,
"source_channel": 2
}, {
"output_channel": 3,
"source_channel": 3
}, {
"output_channel": 4,
"source_channel": 4
}, {
"output_channel": 5,
"source_channel": 5
}, {
"output_channel": 6,
"source_channel": 6
}, {
"output_channel": 7,
"source_channel": 7
}, {
"output_channel": 8,
"source_channel": 8
}],
"line_out_channels": 2,
"line_out_channels_mapping": [{
"output_channel": 1,
"source_channel": 1
}, {
"output_channel": 2,
                
```

```

"source_channel": 2
}, {
"output_channel": 3,
"source_channel": 3
}, {
"output_channel": 4,
"source_channel": 4
}, {
"output_channel": 5,
"source_channel": 5
}, {
"output_channel": 6,
"source_channel": 6
}, {
"output_channel": 7,
"source_channel": 7
}, {
"output_channel": 8,
"source_channel": 8
}],
"hdcp": 1,
"out_colorspace": 0
}
    
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
output_resolution	"1920x1080"	Y	[STRING]	输出分辨率。 auto/1280x720/1920x1080/3840x2160
hdmi_channels	2	Y	[INT]	HDMI 声道数，2/4/6/8
line_out_channels	2	Y	[INT]	LINE OUT 声道数
output_framerate	0	Y	[INT]	输出帧率设置 (注意，此参数在现阶段版本中暂无实际用处)
hdmi_channels_mapping	[{"output_channel":1,"source_channel":1}, {"output_channel": 2,"source_channel":2}, {"output_channel": 3,"source_channel":3}, {"output_channel": 4,"source_channel":4}, {"output_channel": 5,"source_channel":5}, {"output_channel": 6,"source_channel":6}, {"output_channel": 7,"source_channel":7}, {"output_channel": 8,"source_channel":8}]	Y	[OBJECT]	HDMI 声道映射。 注意： 在给当前接口传递该字段时，务必给足全部 8 个声道的值。有设定的部分按照设定给。没有设定的部分按照默认顺序给值。
line_out_channels_mapping	[{"output_channel":1,"source_channel":1}, {"output_channel": 2,"source_channel":2}, {"output_channel": 3,"source_channel":3}, {"output_channel": 4,"source_channel":4}, {"output_channel": 5,"source_channel":5}, {"output_channel": 6,"source_channel":6}, {"output_channel": 7,"source_channel":7}, {"output_channel": 8,"source_channel":8}]	Y	[OBJECT]	LINE OUT 声道映射。 注意： 在给当前接口传递该字段时，务必给足全部 8 个声道的值。有设定的部分按照设定给。没有设定的部分按照默认顺序给值。

	{"output_channel": 8,"source_channel":8}			
source_channel	1	Y	[INT]	源声道数, 1~8 的整数
output_channel	1	Y	[INT]	输出声道数
hdcp	1	Y	[INT]	是否开启高清数字内容保护是否打开。0 关闭; 1 打开
out_colorspace	0	Y	[INT]	色彩空间。 0: RGB444 1: YUV422 2: YUV444

- 响应参数 (样例)

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

- 响应参数说明
无

2.1.5 设置 NDI 连接方式

- 接口描述
设备当前提供的 NDI 连接方式包括 TCP、RUDP、禁用 Multi-TCP 以及组播
- 请求地址
/api/codec/decode/setConnection
- 请求方法
POST
- 请求参数 (样例)

```
{"ndi_connection": "rudp"}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
ndi_connection	"rudp"	Y	[STRING]	NDI 连接方式 TCP: tcp; RUDP: rudp; 禁用 Multi-TCP: unicast; 组播: multicast

- 响应参数 (样例)

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

- 响应参数说明
无

2.2 发现服务模块

2.2.1 设置发现服务器

- 接口描述
通过此接口将设备连接到发现服务器, 在编码模式下, NDI 源 (本设备) 将会注册到发现服务器; 在解码模式下, NDI 发现将从发现服务器上获取 NDI 源列表。这适合于 NDI 跨网段使用或者您的网络不支持 UDP 组播的场情景下使用。
- 请求地址
/api/codec/discovery/setDiscoveryServer
- 请求方法

POST

- 请求参数（样例）
无
- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
enable	true/false	Y	[BOOLEAN]	设置服务开关
address		Y	[STRING]	设置服务器地址

- 响应参数（样例）

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

- 响应参数说明
无

2.2.2 获取发现服务器

- 接口描述
用于提供给前端使用的接口，获取当前机器保存的发现服务器

- 请求地址
/api/codec/discovery/getDiscoveryServer

- 请求方法

GET

- 请求参数（样例）
无

- 请求参数说明
无

- 响应参数（样例）

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

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
enable	[Boolean]	发现服务器开启状态
address	[STRING]	发现服务器地址

2.2.3 扫描获取 NDI Source 列表

- 接口描述
主页扫描到的当前网段以及在手动添加的地址中所有在线的 NDI Source 列表

- 请求地址
/api/codec/discovery/scan

- 请求方法

GET

- 请求参数（样例）
无

- 请求参数说明
无

- 响应参数（样例）

```
{  
  "result": "ok",
```

```

    "data": [
      {
        "name": "DESKTOP-R6TEOQE (vMix - Output 1)",
        "group": "public",
        "original_url": "192.168.38.238:5961",
        "ip": "192.168.38.238",
        "port": 5961,
        "id": 1,
        "group_name": "public",
        "device_name": "DESKTOP-R6TEOQE",
        "channel_name": "vMix - Output 1",
        "series": "NDI",
        "enable": 1,
        "url": "192.168.38.238:5961"
      }
    ],
    "data_size": 1
}
    
```

● 响应参数说明（data 字段，array 里面的单个元素）

字段名	字段类型	字段描述
name	[STRING]	NDI Source Name。这是 NDI Source 原始的名称。如果您想获得设备名称、通道名称，请参考 device_name 和 channel_name。
group	[STRING]	NDI Source 所在的 NDI Group。如果您得到的 Group 是空字符串""，按照 NDI 的规则，它表示默认 public group。
original_url	[STRING]	NDI Source URL。请注意：目前 URL 的格式为 <IP>:<port>，但是建议您不要认为它就是事实，NDI SDK 并不担保这一点。在使用时，请保留 url 的原始值，不要尝试改变它。
ip	[STRING]	ip 地址
port	[INT]	端口号
id	[INT]	发现列表中的序号
group_name	[STRING]	NDI Source 所在的 NDI Group。如果您得到的 Group 是空字符串""，按照 NDI 的规则，它表示默认 public group。
device_name	[STRING]	从 name 中解析出来的 NDI Device Name，通常它也代表了 NDI Source 的 host name。
channel_name	[STRING]	从 name 中解析出来的 NDI Channel Name
series	[STRING]	NDI 源
enable	[INT]	1: 这个 Preset 有定义 0: 这个 Preset 没有定义 NDI Source；在这种情况下，所有后续的字段都没有意义

url	[STRING]	NDI Source URL。请注意：目前 URL 的格式为 <IP>:<port>，但是建议您不要认为它就是事实，NDI SDK 并不担保这一点。在使用时，请保留 url 的原始值，不要尝试改变它。
-----	----------	--

2.2.4 手动添加源配置

- 接口描述
该接口用于手动添加 NDI 源，是对因为跨域问题无法访问到不同网段 NDI 源的一个补充

- 请求地址
/api/codec/discovery/addManualIpsGroups
- 请求方法
POST
- 请求参数（样例）

```
{
  "groups": ["public"],
  "manuals": ["128.168.38.207"]
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
groups	["public"]	Y	[ARRAY(STRING)]	NDI Group
manuals	["128.168.38.207"]	Y	[ARRAY(STRING)]	媒体源地址列表

- 响应参数（样例）

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

- 响应参数说明
无

2.2.5 获取手动添加的 NDI 源地址

- 接口描述
该接口用于提供给前端获取已经手动配置在设备上的 NDI 源的组列表和地址列表。

- 请求地址
/api/codec/discovery/getManualIpsGroups
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```
{"result": "ok", "data": {"groups": [], "manuals": []}}
```

- 响应参数说明

字段名	字段类型	字段描述
groups	[STRING]	NDI Source 所在的 NDI Group。如果您得到的 Group 是空字符串""，按照 NDI 的规则，它表示默认 public group。

manuals	[STRING]	手动添加的 NDI Source 地址列表
---------	----------	-----------------------

2.2.6 设置解码模式下的发现参数

- 接口描述
允许该设备在解码模式下也能被发现，设置此时的 NDI Group 和 NDI Channel

- 请求地址
/api/codec/discovery/setDiscoveryGroupChannel

- 请求方法

POST

- 请求参数（样例）

```
{
  "group_name": "public",
  "channel_name": "Decoding Channel"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
group_name	"public"	Y	[STRING]	设置解码发现的 NDI Group，默认 "public"
channel_name	"Decoding Channel"	Y	[STRING]	设置被发现时用于识别的通道名称

- 响应参数（样例）

无

- 响应参数说明

无

2.2.7 获取该设备在解码模式下的发现参数

- 接口描述
获取该设备在解码模式下被发现的组、通道名称

- 请求地址
/api/codec/discovery/getDiscoveryGroupChannel

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "group_name": "public",
    "channel_name": "Decoding Channel"
  }
}
```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
groups	[STRING]	NDI Source Name。这是 NDI Source 原始的名称。如果您想获得设备名称、通道名称，请参考 device_name 和 channel_name。
channel_name	[STRING]	设置被发现时用于识别的通道名称

2.3 预设模块

我们的 NDI 解码设备有一个非常实用的功能：Preset。Preset 所表示的意义是：您可以将发现的 NDI Source 添加到由 1~9 所代表的 9 个 Presets 中，这是您的"收藏夹(Favorites)"。当您需要快速选择添加在 Preset 中的 NDI Source 进行解码时，您只需要指定 Preset 的 ID (1~9)即可。

2.3.1 获取预设列表

- 接口描述

获取当前已经添加在预设中的所有预设信息，以列表形式输出。

- 请求地址
/api/codec/preset/get
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```
{
  "result": "ok",
  "data": [
    {
      "id": 1,
      "name": "N60X (Chan1)",
      "url": "192.168.38.61:5961",
      "group": "public",
      "online": "on",
      "warning": "",
      "device_name": "N60X",
      "channel_name": "Chan1",
      "ip": "192.168.38.61",
      "port": 5961,
      "enable": 1
    },
    {
      "id": 2,
      "name": "DESKTOP-R6TEOQE (vMix - Output 1)",
      "url": "192.168.38.238:5961",
      "group": "public",
      "online": "on",
```

```

        "warning": "",
        "device_name": "DESKTOP-R6TEOQE",
        "channel_name": "vMix - Output 1",
        "ip": "192.168.38.238",
        "port": 5961,
        "enable": 1
    },
}
    
```

- 响应参数说明（data 字段中，单个元素说明）

字段名	字段类型	字段描述
id	[INT]	1 ~ 9 代表常规的 NDI Source Preset; 0 代表 Blank
name	[STRING]	Preset 所定义的 NDI Source 的名称
url	[STRING]	NDI Source 的原始 URL
group	[STRING]	Preset 所定义的 NDI Source 的 Group name
online	[STRING]	根据网络 NDI 发现的结果，检测出这个 Preset 所定义的 NDI Source 的在线状态: on: 当前 NDI Source 在线（可被发现） off: 当前 NDI Source 不在线（不能发现） 请注意： online 状态是由 NDI Discovery 的结果而决定的，并不代表 NDI Source 是否可以连接和获取视频数据。所以，它只提供参考意义，不能作为设备确定在线的依据。
warning	[STRING]	用于显示当前预设所代表的 NDI Source 是否离线
device_name	[STRING]	从 `name` 字段中提取出来的 NDI Source 的 Device Name
channel_name	[STRING]	从 `name` 字段中提取出来的 NDI Source 的 Channel Name
ip	[STRING]	从 `url` 字段中提出来的 NDI Source 的 IP 地址
port	[INT]	源端口号
enable	[INT]	1: 这个 Preset 有定义 0: 这个 Preset 没有定义 NDI Source; 在这种情况下，所有后续的字段都没有意义

2.3.2 添加预设

- 接口描述
通过请求这个接口，可以将一个有效的 NDI 源添加在相应的位置（1~9）上。
- 请求地址

/api/codec/preset/add

- 请求方法
POST
- 请求参数（样例）

```
{
  "position": 2,
  "name": "N60X (Chan1)",
  "url": "192.168.38.61:5961",
  "group": "public"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
position	2	Y	[INT]	添加预设的位置，一共 1-9 可选
name	"N60X (Chan1)"	Y	[STRING]	NDI Source 的名称，由源决定
url	"192.168.38.61:5961"	Y	[STRING]	NDI Source Url，由源决定
group	"public"	Y	[STRING]	NDI Group，由源决定

- 响应参数（样例）

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

- 响应参数说明
无

2.3.3 删除预设

- 接口描述
可以通过请求该接口，删除参数所对应保存位置的预设。

- 请求地址
/api/codec/preset/remove

- 请求方法
POST
- 请求参数（样例）

```
{"id": 2}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
id	2	Y	[INT]	指定要移除的 Preset ID，有效范围 1~9

- 响应参数（样例）

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

- 响应参数说明
无

2.3.4 设置背景颜色

- 接口描述
此接口设置的是预设中 blank 页面下的解码输出显示颜色

- 请求地址
/api/codec/preset/set_blank_color

- 请求方法

POST

- 请求参数（样例）

```
{"BlankColor": "#ffffff"}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
BlankColor	"#ffffff"	Y	[STRING]	指定要移除的 Preset ID，有效范围 1~9

- 响应参数（样例）

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

- 响应参数说明

无

2.4 解码 PTZ 相机控制

解码 PTZ 相机的意思是，如果您的设备正在解码一台 PTZ 相机编码出的 NDI 源，或者是 PTZ 相机作为源，接入在一台 NDI 编码设备上，并由您的设备正在解码该 NDI 设备的编码源时，您可以在设备的解码模式后台，通过操作 PTZ 控制面板，对编码端的 PTZ 相机进行相应的方位、对焦、缩放等进行控制。

2.4.1 解码模式 PTZ 控制

- 接口描述

解码模式下的 PTZ 控制需要按照对应的参数传入对应的值

- 请求地址

/api/codec/ptz_decode/ptzControl

- 请求方法

GET

- 请求参数（样例）

```
action:"up"
```

```
speed:50
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
action	"up"	Y	[INT]	从下列字符串中选则填写："up"、"left"、"down"、"right"
speed	50	N	[INT]	表示要传入的速度，填入数字 0~100，默认 50 ，像 home、stop-move、stop-all 等等这种可以不用传该值
id	2	N	[INT]	表示传入的位置，主要用于带 preset 结尾的指令，传入数字即可，建议传 1-16
language	"cn"	N	[STRING]	请求头参数，目前版本只支持中英文语言，分别为 "en"/"cn"，用于弹窗提示的多语言输出。 注意： 当前版本中，如果单独请求此接口时想要返

回正确的错误提示，请务必在请求头中加入此参数。否则返回的提示中 msg 字段将为 null

- 响应参数（样例）

```

{"result": "ok"}
{"result": "error", "msg": "无效 PTZ 动作参数"}
{"result": "error", "msg": "PTZ 参数设置不正确"}
    
```

- 响应参数说明
无

2.5 编码

2.5.1 获取所有编码对象名称

- 接口描述
编码对象指的是系统内部用于控制码流的对象，不提供给前端，只在后台自己使用。目前存在三种。

- 请求地址
/api/codec/encoders/list

- 请求方法
POST

- 请求参数（样例）
无

- 请求参数说明
无

- 响应参数

```

[
  "main",
  "sub",
  "main_full"
]
    
```

- 响应参数说明

字段名	字段类型	字段描述
main	[STRING]	NDI HX 主码流对象
sub	[STRING]	NDI HX 子码流对象
main_full	[STRING]	NDI HB 主码流和子码流共同的对象

2.5.2 选择输出的编码模式

- 接口描述
通过请求该接口，可以自由选择编码 NDI|HX 编码还是 NDI|HB 编码，以及网络流模式，抑或是某两种的组合模式。

- 请求地址
/api/codec/encoders/choose_encode

- 请求方法
POST

- 请求参数（样例）

```
{"main":1,"main_full":1,"srt":0}
```

- 请求参数说明

参数名称	参数值	是否必填	参数类型	参数描述
main	1	Y	[INT]	NDI HX 流 0: 关闭 1: 开启
main_full	1	Y	[INT]	NDI HB 流 0: 关闭 1: 开启
srt	0	Y	[INT]	多协议 srt 流 0: 关闭 1: 开启

- 响应参数（样例）

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

- 响应参数说明
无

2.5.3 获取码流编码参数

- 接口描述

通过向该接口路径中编码中的 {encoder_stream} 传入指定的对象，则可以获得当前编码对象下码流的编码参数设置情况

- 请求地址
`/api/codec/encoder/{encoder_stream}/get`
- 请求方法
POST
- 请求参数（样例）
无
- 请求参数说明

参数名称	参数值	是否必填	参数类型	参数描述
encoder_stream	"main"/"main_full"	Y	[STRING]	路径参数 main: NDI HX 主码流 sub: NDI HX 子码流; main_full: NDI HB 主码流 注意: 正常情况下可以填写上述三个值，但是最好不要填写 sub，因为当前版本下，子码流获取值的参数并不准确。

- 响应参数（样例）

```
{
  "yuv": "NV12",
  "max_bitrate": 10000,
  "gdr_mode": 0,
  "gop_length": 60,
  "b_frames": 0,
  "target_bitrate": 10000,
  "cpb_size": 1000,
  "initial_delay": 500,
  "periodicity_idr": 60,
  "prefetch_buffer": true,
```

```

"low_bandwidth": false,
"filler_data": true,
"control_rate": "cbr",
"hx_version": 2,
"mode": "x-h264",
"profile": "Main",
"alignment": "nal",
"mode_version": 1,
"started": true
}
    
```

- 响应参数说明

字段名	字段类型	字段描述
yuv	[STRING]	YUV 颜色编码格式
max_bitrate	[INT]	输出视频流的最大比特率
gdr_mode	[INT]	渐近解码器 (Gradual Decoder Refresh) 刷新方案
gop_length	[INT]	一组连续图像帧中包含的帧数
b_frames	[INT]	设置 B 帧的数量
target_bitrate	[INT]	码率
cpb_size	[INT]	象在 HRD 模型中指定的编码图像缓存 (CPB)
initial_delay	[INT]	象在 HRD 模型中指定的初始删除延迟 (以毫秒为单位)
periodicity_idr	[INT]	指定 IDR 帧的周期性
prefetch_buffer	[BOOLEAN]	开启/关闭编码过程中的 L2Cache 缓冲区
low_bandwidth	[BOOLEAN]	指定低带宽模式。减少用于 P 帧运动估计的垂直搜索范围
filler_data	[BOOLEAN]	在 CBR 速率控制模式下启用/禁用填充数据添加功能
control_rate	[STRING]	码率控制方式。low-latency: 低延时; cbr: CBR; vbr: VBR
hx_version	[INT]	HX 版本。2: HX2; 3: HX3
mode	[STRING]	x-h264/x-h265 编码格式
profile	[STRING]	编码画质级别。可用 Baseline, Extended, Main, High
alignment	[STRING]	时延模式, au/nal
mode_version	[INT]	1: x-h264 2: x-h265
started	[BOOLEAN]	码流开启与否

2.5.4 获取视频源参数

- 接口描述

通过向该接口路径中编码中的 {encoder_stream} 传入指定的对象, 获取当前编码对象在视频源层面的基本参数, 包括分辨率、信号, 编码对象是否激活、编码流状态等

- 请求地址

/api/codec/encoder/{encoder_stream}/get_capture

- 请求方法

POST

- 请求参数（样例）
无
- 请求参数说明

参数名称	参数值	是否必填	参数类型	参数描述
encoder_stream	"main"/"main_full"	Y	[STRING]	main: 主码流 main_full: FULL NDI 主码流 multi_protocol: 多协议码流

- 响应参数（样例）

```
{
  "width": 1920,
  "height": 1080,
  "framerate": 60,
  "interlaced": 0,
  "io_mode": 4,
  "device": "/dev/video0",
  "framerate_D": 1000,
  "enable": 1,
  "started": true,
  "signal": "none",
  "resolution": "1920x1080p 60Hz",
  "source": "sdi"
}
```

- 响应参数说明

字段名	字段类型	字段描述
width	[INT]	视频宽
height	[INT]	视频高
framerate	[INT]	帧率
interlaced	[INT]	交织模式
io_mode	[INT]	视频源读取模式
device	[STRING]	设备节点
framerate_D	[INT]	帧率分母
enable	[INT]	当前码流是否开启的开关
started	[BOOLEAN]	码流管道是否开启
signal	[STRING]	是否有音视频信号
resolution	[STRING]	当前格式
source	[STRING]	视频源接入方式。sdi、hdmi

2.5.5 设置主码流（NDI|HX）编码参数

- 接口描述

通过请求该接口对 NDI|HX 的编码流进行编码参数设置。在当前版本中，部分参数（如 target_bitrate）设置后，可以实时生效，而部分参数（如 mode_version）设置后，编码端会重启编码，出现的效果就是黑屏一小会。

- 请求地址
/api/codec/encoder/main/set
- 请求方法

POST

- 请求参数（样例）

```
{
  "target_bitrate":15000,
  "key":"main"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
key	"main"	Y	[STRING]	指定当前需要调整的码流，main 和 main_full。 注意：需要和当前的 url 路径匹配。
target_bitrate	6000	N	[INT]	编码质量（码率） 2000: 极低 6000: 低 10000: 默认 15000: 高 20000: 极高
mode_version	1	N	[INT]	1: x-h264 2: x-h265
hx_version	2	N	[INT]	2: NDI HX2 3: NDI HX3
yuv	"NV12"	N	[STRING]	色彩空间： NV12: YUV420 8bit NV16: YUV420 10bit NV12_10LE32: YUV422 8bit NV16_10LE32: YUV422 10bit

- 响应参数（样例）

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

- 响应参数说明

无

2.5.6 设置 FULL NDI 流（NDI|HB）编码参数

- 接口描述

通过请求该接口对 NDI|HB 的编码流进行编码参数设置。

- 请求地址

/api/codec/encoder/main_full/set

- 请求方法

POST

- 请求参数（样例）

```
{
  "target_bitrate":115,
  "key":"main_full"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
------	-------	------	------	------

key	"main"	Y	[STRING]	指定当前需要调整的码流，main 和 main_full。 注意：需要和当前的 url 路径匹配。
target_bitrate	100	N	[INT]	编码质量（码率） 55: 极低 75: 低 100: 默认 150: 高 200: 极高

- 响应参数（样例）

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

- 响应参数说明
无

2.5.7 设置多协议流编码参数

- 接口描述
通过请求该接口对多协议的编码流进行编码参数设置。

- 请求地址
`/api/codec/encoder/multi_protocol/set`

- 请求方法

POST

- 请求参数（样例）

```
{
  "target_bitrate":115,
  "key":"multi_protocol"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
key	"multi_protocol"	Y	[STRING]	指定当前需要调整的码流，multi_protocol。 注意：需要和当前的 url 路径匹配。
target_bitrate	100	N	[INT]	编码质量（码率） 55: 极低 75: 低 100: 默认 150: 高 200: 极高

- 响应参数（样例）

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

- 响应参数说明
无

2.5.8 获取流信息

- 接口描述
通过向该接口路径中编码中的 {encoder_stream} 传入指定的对象，从而获取该对象的流配置信息。

- 请求地址
`/api/codec/streams/{stream_name}/list`

- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明

参数名称	参数值	是否必填	参数类型	参数描述
stream_name	"main"/"main_full"	Y	[STRING]	main: NDI HX 的推流, main_full: NDI HB 的推流, multi_protocol: 多协议流 路径参数

- 响应参数（样例）

```
// NDI-HX 流
[
  {
    "type": "NDI-HX",
    "name": "ndi-hx",
    "enable": true,
    "group": "1212222",
    "ttl": "127",
    "channel_name": "Chan1",
    "connection": "disable_rudp",
    "netmask": "255.255.0.0",
    "netprefix": "239.255.0.0",
    "discovery_server": {
      "enable": false,
      "address": ""
    },
    "location": null
  }
]

// NDI-HB 流
[
  {
    "group": "",
    "ttl": 127,
    "channel_name": "FULL-24844",
    "connection": "rudp",
    "netmask": "255.255.0.0",
    "netprefix": "239.255.0.0",
    "discovery_server": {
      "enable": true,
      "address": "192.168.4.66"
    }
  }
]
```

```
}
]
```

- 响应参数说明

字段名	字段类型	字段描述
type	[STRING]	流类型。NDI-HX、NDI-HB、SRT
name	[STRING]	流名称
enable	[BOOLEAN]	当前流是否开启
group	[STRING]	组名
channel_name	[STRING]	NDI 通道名称
ttl	[INT]	存活时间
connection	[STRING]	连接方式
netmask	[STRING]	子网掩码
netprefix	[STRING]	组播地址
discovery_server	[OBJECT]	是否连接到发现服务器。 address: 发现服务器地址 enable: 是否连接到发现服务器

2.5.9 所有流重新推流

- 接口描述

通过请求此接口，可以将 HX 和 HB 的码流均重新推流，路径参数可以从“main”或者“main_full”中任选其一填入即可。

- 请求地址

`/api/codec/streams/{stream_name}/reset`

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

参数名称	参数值	是否必填	参数类型	参数描述
stream_name	"main"/"main_full"	Y	[STRING]	路径参数： main main_full multi_protocol 指定流名称

- 响应参数（样例）

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

- 响应参数说明

无

2.5.10 获取流相关属性及配置

- 接口描述

获取 NDI|HX 和 NDI|HB 的相关流的属性配置。

- 请求地址

`/api/codec/streams/{stream_name}/{stream_key}/get`

- 请求方法

GET

- 请求参数（样例）
无
- 请求路径参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
stream_name	"main"	Y	[STRING]	路径参数值 main: NDI HX 流； main_full: NDI HB 流； multi_protocol: 多协议网络流。
stream_key	"ndi-hx"	Y	[STRING]	路径参数值 ndi-hx: 该参数值与 stream_name 参数值 main 对应； ndi-full: 该参数与 stream_name 参数值 main_full 对应； srt: 该参数与 stream_name 参数值 multi_protocol 对应。 注意: 当 name 为 "main" 时, key 必须填写 "ndi-hx"; 当 name 为 "main_full" 时, key 必须填写 "ndi-full"; 当 name 为 "multi_protocol" 时, key 必须填写 "srt"。

- 响应参数（样例）

```
{
  "type": "NDI-HX",
  "name": "ndi-hx",
  "enable": true,
  "group": "1212222",
  "ttl": "127",
  "channel_name": "Chan1",
  "connection": "disable_rudp",
  "netmask": "255.255.0.0",
  "netprefix": "239.255.0.0",
  "discovery_server": {
    "enable": false,
    "address": ""
  },
  "location": null,
  "started": true,
  "source": "sdi"
}
```

- 响应参数说明

字段名	字段类型	字段描述
type	[STRING]	流类型: NDI-HX、NDI-HB、SRT
name	[STRING]	流名称
enable	[BOOLEAN]	当前流是否开启
group	[STRING]	组名
channel_name	[STRING]	NDI 通道名称
ttl	[INT]	存活时间
connection	[STRING]	连接方式
netmask	[STRING]	子网掩码
netprefix	[STRING]	组播地址

discovery_server	[OBJECT]	是否连接到发现服务器。 address: 发现服务器地址 enable: 是否连接到发现服务器
location	[STRING]	(当前版本废弃字段)
started	[BOOLEAN]	流是否开启
source	[STRING]	流接入来源。sdi、hdmi。N60 写死为 hdmi

2.5.11 获取流实时码率、连接数

- 接口描述
页面上显示得码率是由系统后台发送给前端页面的，而本接口则是提供了一个接口用于前端主动获取当前的码率和连接数。

- 请求地址
/api/codec/streams/{stream_name}/{stream_key}/get_dynamic_state
- 请求方法
GET
- 请求参数 (样例)
无
- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
stream_name	"main"	Y	[STRING]	路径参数值 main: NDI HX 流; main_full: NDI HB 流; multi_protocol: 多协议网络流。
stream_key	"ndi-hx"	Y	[STRING]	路径参数值 ndi-hx: 该参数值与 stream_name 参数值 main 对应; ndi-full: 该参数与 stream_name 参数值 main_full 对应; srt: 该参数与 stream_name 参数值 multi_protocol 对应。 注意: 当 name 为 "main" 时, key 必须填写 "ndi-hx"; 当 name 为 "main_full" 时, key 必须填写 "ndi-full"; 当 name 为 "multi_protocol" 时, key 必须填写 "srt"。

- 响应参数 (样例)

```
{
  "bitrate": 0,
  "connections": 0
}
```

- 响应参数说明

字段名	字段类型	字段描述
bitrate	[INT]	此流实时码率
discovery_connections	[INT]	此流实时连接数。 注意: 当前设备版本不支持显示流实时连接数, 此字段为保留字段, 默认为 0

2.5.12 更新流相关配置

- 接口描述
通过该接口可以调整当前流的 NDI Group、NDI channel，以及 NDI 连接方式等参数。

- 请求地址
`/api/codec/streams/{stream_name}/{stream_key}/set`

- 请求方法
POST
- 请求参数（样例）

```
{
  "group": "",
  "channel_name": "Chan1-248",
  "connection": "rudp",
  "types": "ndi-hx"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
stream_name	"main"	Y	[STRING]	路径参数值 main: NDI HX 流; main_full: NDI HB 流; multi_protocol: 多协议网络流。
stream_key	"ndi-hx"	Y	[STRING]	路径参数值 ndi-hx: 该参数值与 stream_name 参数值 main 对应; ndi-full: 该参数与 stream_name 参数值 main_full 对应; srt: 该参数与 stream_name 参数值 multi_protocol 对应。 注意: 当 name 为"main"时, key 必须填写"ndi-hx"; 当 name 为"main_full"时, key 必须填写 "ndi-full"; 当 name 为 "multi_protocol"时, key 必须填写"srt"。
group	"public"	N	[STRING]	NDI Group
channel_name	"Chan1-248"	N	[STRING]	NDI 通道名称
connection	"disable_rudp"	N	[STRING]	连接方式: disable_rudp: 默认 rudp: RUDP udp: 禁用 Multi-TCP multicast: 组播
types	"ndi-hx"	N	[STRING]	与 stream_key 参数值保持一致

- 响应参数（样例）

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

- 响应参数说明
无

2.6 模式

N60 同时具备 Encoder 和 Decoder 的能力，您可以在这两者之间进行切换。

2.6.1 设置工作模式（编码模式 | 解码模式）

- 接口描述

通过此接口，您可以自由地在编码模式和解码模式之间进行切换。注意，切换后，设备将会重启并加载新的资源，会有 1 分钟左右的等待时间。

- 请求地址

/api/codec/mode/set?mode=encode

- 请求方法

POST

- 请求参数（样例）

无

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
mode	"encode"/"decode"	Y	[STRING]	encode: 编码模式; decode: 解码模式

- 响应参数（样例）

true

- 响应参数说明

无

2.6.2 获取工作模式

- 接口描述

通过请求此接口，可以直接获得当前设备所处的编解码模式。encode: 编码模式；decode: 解码模式

- 请求地址

/api/codec/mode/get

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

encode

- 响应参数说明

字段名	字段类型	字段描述
-	[TEXT]	encode: 编码模式; decode: 解码模式

2.7 编码 PTZ 控制

2.7.1 编码 PTZ 控制

- 接口描述

通过请求本接口，传入相应的动作和速度等，模拟页面上通过面板操控相机的方式，可以直接控制 PTZ 的方向、对焦和缩放。

- 请求地址

/api/codec/ptz/ptzControl

- 请求方法

GET

- 请求参数（样例）

```
action:"up"
```

```
speed:50
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
action	"up"	Y	[INT]	从下列字符串中选则填写："up"、"left"、"down"、"right"
speed	50	N	[INT]	表示要传入的速度，填入数字 0~100，默认 50 ，像 home、stop-move、stop-all 等等这种可以不用传该值
id	2	N	[INT]	表示传入的位置，主要用于带 preset 结尾的指令，传入数字即可，建议传 1-16
language	"cn"	N	[STRING]	请求头参数，目前版本只支持中英文语言，分别为 "en"/"cn"，用于弹窗提示的多语言输出。 注意： 当前版本中，如果单独请求此接口时想要返回正确的错误提示，请务必在请求头中加入此参数。否则返回的提示中 msg 字段将为 null

- 响应参数（样例）

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

- 响应参数说明

无

2.7.2 获取 PTZ 信息

- 接口描述

该接口主要是获取当前设备上对于 PTZ 相机的连接和控制的配置信息。

- 请求地址

/api/codec/ptz/getPtzInfo

- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "enable": 1,
    "typ": "network",
    "network": {
      "protocol": "TCP",
      "addr": "192.168.38.18",
      "port": 50381,
      "ptz_protocol": "Sony Visca",
      "ptz_addr": "1"
    },
    "serial": {
      "device": "none",
      "baudrate": 9600,
      "parity": "none",
      "startBits": 8,
      "endBits": 1,
      "xonxoff": 0,
      "rtscts": 0,
      "ptz_protocol": "Sony Visca",
      "ptz_addr": "1"
    }
  }
}
```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
enable	[BOOLEAN]	PTZ 功能是否开启
typ	[STRING]	连接方式 network: 网络连接 serial: 串口连接
network	[OBJECT]	网络连接 protocol: 网络协议。TCP/UDP; addr: PTZ 相机 IP 地址; port: PTZ 相机端口; ptz_protocol: PTZ 协议。"Sony ; Visca"、"Pelco-d" 和 "Pelco-p"
serial	[OBJECT]	串口连接 device: 串口驱动; baudrate: 波特率; parity: 校验; startBits: 数据位; xonxoff: XON/XOFF 开关; rtscts: RTS/CTS 开关

2.7.3 获取 PTZ 协议

- 接口描述

当前版本下，N60 所支持的协议包括 Sony Visca、Pelco-d、Pelco-p，通过请求该接口可以通过返回字段的 data 字段获取到相应的协议名称。

- 请求地址

/api/codec/ptz/getPtzProtocol

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

```
{
  "result": "ok",
  "msg": null,
  "data": [
    "Sony Visca",
    "Pelco-d",
    "Pelco-p"
  ]
}
```

- 响应参数说明

无

2.7.4 获取串口设备

- 接口描述

当有串口设备接入，或者直接使用串口线插入本机器，则可以通过此接口，实现获取到串口设备的设备节点功能

- 请求地址

/api/codec/ptz/getSerialDevice

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

```
{
  "result": "ok",
  "data": [
    {
      "device": "ttyUSB0",
      "driver": "FTDI"
    }
  ]
}
```

- 响应参数说明

无

2.7.5 修改 PTZ 信息

- 接口描述
通过本接口，传入相应的字段参数，修改设备上保存的 PTZ 连接和控制信息

- 请求地址
/api/codec/ptz/modify

- 请求方法
POST

- 请求参数（样例）

```
{
"enable": 1,
"typ": "network",
"device": "none",
"baudrate": 9600,
"parity": "none",
"startBits": 8,
"endBits": 1,
"xonxoff": 0,
"rtscts": 0,
"ptz_protocol": "Sony Visca",
"ptz_addr": 1,
"protocol": "UDP",
"addr": "192.168.38.18",
"port": 50381
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
enable	0	Y	[INT]	ptz 控制开关。 0 关闭；1 开启
typ	"network"	Y	[STRING]	ptz 类型。 network/serial
device	"ttyUSB0"	N	[STRING]	串口驱动设备
baudrate	9600	N	[INT]	波特率
parity		N	[STRING]	奇偶校验。
startBits	8	N	[INT]	数据位
endBits	1	N	[INT]	停止位
xonxoff	0	N	[INT]	XON/XOFF 开关
rtscts	0	N	[INT]	RTS/CTS 开关
ptz_protocol	"Sony Visca"	N	[STRING]	PTZ 协议，目前可以选填的协议包括如下： "Sony Visca", "Pelco-d", "Pelco-p"
ptz_addr	1	N	[INT]	串口地址位置。协议内部的固定参数。 数字，1 或者 0
protocol	UDP/TCP	N	[STRING]	网络协议

addr	"192.168.38.18"	N	[STRING]	ptz 相机 ip 地址
port	50381	N	[INT]	ptz 相机端口

- 响应参数（样例）

```
{"result": "ok", "msg": null}
```

- 响应参数说明
无

2.8 二维码扫描

2.8.1 虚拟用户身份登录

- 接口描述
扫描二维码的前端第一个请求，用于生成一个虚拟用户 ID，负责后续记录用户的登录状态和唯一识别。

- 请求地址
/api/codec/qr_code/virtual_login

- 请求方法

GET

- 请求参数（样例）
无

- 请求参数说明
无

- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "current_virtual_user": "1e1ffd2ec06c43e6a9efe6022162ac74"
  }
}
```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
current_virtual_user	[STRING]	虚拟登录用户的标识 ID

2.8.2 获取二维码的 ID

- 接口描述
扫描二维码的前端第二个请求，用于获取小屏面板的二维码唯一标识 id，为后续的校验做准备

- 请求地址
/api/codec/qr_code/qrcode_get

- 请求方法

GET

- 请求参数（样例）
无

- 请求参数说明
无

- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "qrcode_id": "84598cea1a878c6f2b7e3db89977b0e2"
  }
}
```

```
}
}
```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
qrcode_id	[STRING]	二维码的标识 ID

2.8.3 二维码校验

- 接口描述

同一个码只允许一个人扫码后进入解码页面，允许多个已经扫过码的同时在线（刷新不影响）。

- 请求地址

/api/codec/qr_code/qrcode_session

- 请求方法

POST

- 请求参数（样例）

```
{
"current_virtual_user": "1e1ffd2ec06c43e6a9efe6022162ac74",
"qrcode_id": "84598cea1a878c6f2b7e3db89977b0e2",
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
current_virtual_user	"1e1ffd2ec06c43e6a9efe6022162ac74"	Y	[STRING]	虚拟用户 token
qrcode_id	"84598cea1a878c6f2b7e3db89977b0e2"	N	[INT]	二维码 id
language	"cn"	N	[STRING]	请求头参数，当前版本只支持中英文语言，用于在返回错误结果时给出弹窗提示的多语言输出。 注意： 如果不提示，那么在出现错误时，返回的字段 msg 的值将为空

- 响应参数（样例）

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

- 响应参数说明

无

2.8.4 检查当前是否为解码模式

- 接口描述

此方法用于给后端提供一个检查当前编解码模式的方法，仅做保留。目前检查编解码模式的工作放在了前端处理。

- 请求地址
/api/codec/qr_code/check_mode
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
language	"en"	N	[STRING]	请求头参数，当前版本只支持中英文语言，用于在返回错误结果时给出弹窗提示的多语言输出。 注意： 如果不提示，那么在出现错误时，返回的字段 msg 的值将为空

- 响应参数（样例）

解码模式下响应结果：

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

编码模式下响应结果（当 language 字段值为"en"）：

```
{
  "result": "verificationError",
  "msg": "Please set the device to decoding mode and re-scan the QR!"
}
```

- 响应参数说明
无

2.8.5 获取二维码

- 接口描述
手机端点击二维码小图标，将可以显示出和小屏面板上二维码页面一样的二维码。

- 请求地址
/api/codec/qr_code/get_qrcode
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "image_path":
"192.168.38.217/api/codec/qrcode_img/84598cea1a878c6f2b7e3db89977b0e2_PC.png"
  }
}
```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
image_path	[STRING]	获取机器上保存的用于给 PC 或者手机端显示的当前二维码图片

2.9 页面预览画面（WebRTC）

2.9.1 获取 webrtc 的 sdp 和 port

- 接口描述
此接口可以用于获取 webrtc 的 sdp 具体内容和端口号

/api/codec/webrtc/add_webrtc

- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```
{
  "port": 60620,
  "sdp": "v=0\r\no=- 1495799811084970 1495799811084970 IN IP4 192.168.40.232\r\ns=-\r\nnt=0
0\r\na=group:BUNDLE video\r\na=msid-semantic: WMS \r\na=ice-lite\r\nm=video 1 RTP/SAVPF 126\r\nnc=IN IP4
0.0.0.0\r\na=mid:video\r\na=sendonly\r\na=rtcp-mux\r\na=ice-ufraq:5bkq\r\na=ice-pwd:0yxuvkzbnakr6k1rl7u78
548\r\na=ice-options:trickle\r\na=fingerprint:sha-256
DE:2B:B8:5D:19:DE:1A:13:78:1C:57:E2:30:D8:B2:0F:B3:FB:18:71:02:7D:6B:80:EB:24:FA:9E:6A:3D:CD:0A\r\na=setu
p:actpass\r\na=rtpmap:126 H264/90000\r\na=fmtp:126
level-asymmetry-allowed=1;packetization-mode=1;profile-level-id=42e01f\r\na=candidate:1 1 udp 12345678
192.168.40.232 60620 typ host\r\n"
}
```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
port	[INT]	媒体流的端口
sdp	[STRING]	获取到的多媒体会话的协议的具体内容

2.9.2 设置 webrtc 的模式

- 接口描述
此接口可以用于获取 webrtc 的 sdp 具体内容和端口号

/api/codec/webrtc/set_webrtc

- 请求方法
GET
- 请求参数（样例）

字段名	字段类型	字段描述
port	[INT]	媒体流的端口
mode	[STRING]	Webrtc 的模式

- 请求参数说明
无
- 响应参数（样例）

- 无
- 响应参数说明（data 字段说明）
无

3 网络

3.1 网络管理

3.1.1 获取全部网络信息

- 接口描述
该接口将返回两个网卡下的所有网络信息设置情况
- 请求地址
/api/networkmanager/network/GetLinkinfo
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```
{
  "result": "ok",
  "data": [
    {
      "status": "up",
      "address": "192.168.38.217",
      "netmask": "255.255.255.0",
      "gw": "192.168.38.254",
      "mac": "56:3D:69:9B:5A:C7",
      "method": "static",
      "enable": true,
      "dns": "223.5.5.5;224.5.5.5",
      "device": "eth0"
    },
    {
      "status": "down",
      "address": "0.0.0.0",
      "netmask": "255.255.255.0",
      "gw": "",
      "mac": "44:07:6A:5F:D4:F7",
      "method": "dhcp",
      "enable": true,
      "dns": "223.5.5.5;224.5.5.5",
      "device": "eth1"
    }
  ]
}
```

- 响应参数说明（data 字段中单个元素说明）

字段名	字段类型	字段描述
-----	------	------

status	[STRING]	当前网络是否被激活 状态由当前网口是否接入网络决定。 "up": 网络激活 "down": 网络未激活
address	[STRING]	IP 地址。可能来自 dhcp 或者手动设置。
netmask	[STRING]	子网掩码
gw	[STRING]	网关
mac	[STRING]	MAC 地址
method	[STRING]	地址获取方式 dhcp: DHCP 自动获取; static: 手动设置
enable	[BOOLEAN]	当前网卡是否有效。当前版本该字段值默认为 true
dns	[STRING]	DNS, 每个 dns 地址由分号隔开, 返回的是字符串。
device	[STRING]	网卡

3.1.2 修改当前设备网络信息

- 接口描述
通过请求该接口修改当前设备的网络信息。备用接口, 实际在本机器中并未使用该接口设置网络信息。

- 请求地址
/api/networkmanager/network/ChangeLinkinfo

- 请求方法
POST

- 请求参数 (样例)

```
{
  "ifname": "eth0",
  "cfg": {
    "address": "0.0.0.0",
    "netmask": "255.255.255.0",
    "gw": "",
    "mac": "44:07:6A:5F:D4:F7",
    "method": "dhcp",
    "enable": true,
    "dns": "223.5.5.5;224.5.5.5",
  }
}
```

- 请求参数说明

参数名称	参数值样板	是否必填	参数类型	参数描述
address	"192.168.38.16"	Y	[STRING]	请求头参数, 当前版本只支持中英文语言, 用于在返回错误结果时给出弹窗提示的多语言输出。 注意: 如果不提示, 那么在出现错误

				时,返回的字段 msg 的值将为空
netmask	"255.255.255.0"	Y	[STRING]	子网掩码
gw	"192.168.38.254"	Y	[STRING]	网关
mac	"44:07:6A:5F:D4:F7"	Y	[STRING]	MAC 地址
method	"dhcp"	Y	[STRING]	地址获取方式 dhcp: DHCP 自动获取; static: 手动设置
enable	true	Y	[BOOLEAN]	当前网卡是否有效。手动设置
dns	"223.5.5.5;224.5.5.5"	N	[STRING]	DNS, 每个 dns 地址由分号隔开, 返回的是字符串。

- 响应参数 (样例)

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

- 响应参数说明
无

3.1.3 以太网开关

- 接口描述

用于控制当前的网卡是否激活。

- 请求地址

/api/networkmanager/network/SetSwitch

- 请求方法

POST

- 请求参数 (样例)

```
{
  "ifname": "eth0",
  "enable": true
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
ifname	"eth0"	Y	[STRING]	网卡
enable	true	Y	[BOOLEAN]	开启/关闭当前网卡

- 响应参数 (样例)

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

- 响应参数说明
无

3.1.4 获取网卡信息

- 接口描述

依据传入的网卡名称来获取相应的信息

- 请求地址

/api/networkmanager/network/GetEthernets

example:

/api/networkmanager/network/GetEthernets?ifname=eth0

- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明

参数名称	参数值	是否必填	参数类型	参数描述
ifname	"eth0"/"eth1"	Y	[STRING]	网卡

- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "status": "up",
    "address": "192.168.38.217",
    "netmask": "255.255.255.0",
    "gw": "192.168.38.254",
    "mac": "56:3D:69:9B:5A:C7",
    "method": "static",
    "enable": true,
    "dns": "223.5.5.5;224.5.5.5",
    "device": "eth0"
  }
}
```

- 响应参数说明（data 字段说明）

字段名	字段类型	字段描述
status	[STRING]	当前网络是否被激活 状态由当前网口是否接入网络决定。 "up": 网络激活 "down": 网络未激活
address	[STRING]	IP
netmask	[STRING]	子网掩码
gw	[STRING]	网关
mac	[STRING]	MAC 地址
method	[STRING]	地址获取方式 dhcp: DHCP 自动获取; static: 手动设置
enable	[BOOLEAN]	当前网卡是否开启。当前版本该字段值默认为 true
dns	[STRING]	DNS, 每个 dns 地址由分号隔开, 返回的是字符串。
device	[STRING]	网卡

3.1.5 设置网卡信息

- 接口描述
该接口为设备上正在使用的用于设置网卡的信息
- 请求地址
/api/networkmanager/network/SetEthernets

- 请求方法
POST
- 请求参数（样例）

```
{
  "ifname": "eth0",
  "address": "192.168.38.217",
  "netmask": "255.255.255.0",
  "gw": "192.168.38.254",
  "mac": "56:3D:69:9B:5A:C7",
  "method": "static",
  "dns": "223.5.5.5;224.5.5.5",
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
ifname	"eth0"	Y	[STRING]	网卡。"eth0"/"eth1"
address	"192.168.38.217"	Y	[STRING]	IP
netmask	"255.255.255.0"	Y	[STRING]	网络掩码
gw	"192.168.38.254"	Y	[STRING]	网关
mac	"56:3D:69:9B:5A:C7"	Y	[STRING]	MAC 地址
method	"static"	Y	[STRING]	地址获取方式 dhcp: DHCP 自动获取; static: 手动设置
dns	"223.5.5.5;224.5.5.5"	Y	[STRING]	DNS 服务器

- 响应参数（样例）

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

- 响应参数说明
无

3.1.6 设置链路聚合

- 接口描述
本接口用于激活和设置链路聚合模式。设置链路聚合模式后，普通网卡模式将失效

- 请求地址
/api/networkmanager/network/SetBonding

- 请求方法

POST

- 请求参数（样例）

```
{
  "mac": "96:82:70:FA:36:98",
  "address": "192.168.43.77",
  "netmask": "255.255.255.0",
  "gw": "192.168.43.254",
  "dns": ["223.5.5.5", "223.6.6.6", "101.226.4.6", "180.76.76.76", "114.114.114.114", "119.29.29.29"],
  "mode": "0",
  "method": "dhcp"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
address	"192.168.43.77"	Y	[STRING]	设备的 IP 地址

netmask	"255.255.255.0"	Y	[STRING]	网络掩码
gw	"192.168.43.254"	Y	[STRING]	网关
mac	"96:82:70:FA:36:98"	Y	[STRING]	MAC 地址
method	"dhcp"	Y	[STRING]	地址获取方式 dhcp: DHCP 自动获取; static: 手动设置
dns	["223.5.5.5", "223.6.6.6", "101.226.4.6", "180.76.76.76", "114.114.114.114", "119.29.29.29"]	Y	[STRING]	DNS 服务器
mode	"0"	Y	[STRING]	聚合模式。 0: 轮询均衡模式; 1: 主备模式

- 响应参数（样例）

该接口请求成功后，设备的 IP 等信息会修改，固接口不会有返回值。（需要返回状态值，建议使用子线程）

- 响应参数说明
无

3.1.7 获取链路聚合信息

- 接口描述

通过请求本接口可以获取链路聚合信息，包括此时的地址、网关、dns 等。

- 请求地址
/api/networkmanager/network/GetBondingInfo
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```
{
  "status": true,
  "desc": "链路聚合已开",
  "method": "dhcp",
  "mac": "56:3D:69:9B:5A:C7",
  "address": "192.168.38.217",
  "netmask": "255.255.255.0",
  "gw": "192.168.38.254",
  "dns": [
    "223.5.5.5",
    "223.6.6.6",
    "101.226.4.6",
    "180.76.76.76",
    "114.114.114.114",
    "119.29.29.29"
  ]
}
```

```

    ],
    "mode": "1",
    "eth0_status": "up",
    "eth1_status": "down"
}
    
```

- 响应参数说明

字段名	字段类型	字段描述
status	[STRING]	如果该字段的值为 false，代表当前的网络为普通模式，不是链路聚合模式；否则，就是链路聚合模式。
desc	[STRING]	IP 地址
method	[STRING]	地址获取方式 dhcp: DHCP 自动获取； static: 手动设置
mac	[STRING]	MAC 地址
address	[STRING]	设备的 IP 地址
netmask	[STRING]	子网掩码
gw	[BOOLEAN]	网关
dns	[STRING]	DNS，每个 dns 地址由分号隔开，返回的是字符串。
mode	[STRING]	聚合模式。 0: 轮询均衡模式； 1: 主备模式
eth0_status	[STRING]	目前网卡 1 的状态
eth1_status	[STRING]	目前网卡 2 的状态

3.1.8 重启 MDNS

- 接口描述

请求当前接口将立即重启 MDNS

- 请求地址

/api/networkmanager/network/ResetMDNS

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

```

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

- 响应参数说明

无

3.2 静态路由管理

3.2.1 添加静态路由

- 接口描述

请求此接口添加静态路由地址

- 请求地址
/api/networkmanager/static/routing/add
- 请求方法
POST
- 请求参数（样例）

```
{
  "interface": "eth0",
  "target": "192.168.38.12",
  "netmask": "255.255.255.0",
  "gateway": "192.168.38.254",
  "mtu": 1500,
  "metric": 0
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
interface	"eth0"	Y	[STRING]	网卡
target	"192.168.42.76"	Y	[STRING]	目标 IP 地址
netmask	"255.255.255.0"	Y	[STRING]	网络掩码
gateway	"192.168.43.254"	Y	[STRING]	网关
metric	0	Y	[INT]	路由开销
mtu	1500	Y	[INT]	最大传输单元

- 响应参数（样例）

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

- 响应参数说明
无

3.2.2 获取静态路由列表

- 接口描述
如果设备中已经有保存过之前设置的静态路由列表，则可以请求此接口获取静态路由列表。

- 请求地址
/api/networkmanager/static/routing/get
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```
{
  "interface": "eth0",
  "target": "192.168.38.12",
  "netmask": "255.255.255.0",
  "gateway": "192.168.38.254",
}
```

```
"mtu":1500,
"metric":0
}
```

- 响应参数说明

字段名	字段类型	字段描述
interface	[STRING]	网卡
target	[STRING]	目标
netmask	[STRING]	网络掩码
gateway	[STRING]	网关
metric	[INT]	路由开销
mtu	[INT]	最大传输单元
status	[INT]	路由启用状态

3.2.3 修改静态路由

- 接口描述

请求此接口，修改指定的静态路由

- 请求地址

/api/networkmanager/static/routing/modify

- 请求方法

POST

- 请求参数（样例）

```
{
  "id":1,
  "interface":"eth0",
  "target":"192.168.38.12",
  "netmask":"255.255.255.0",
  "gateway":"192.168.38.254",
  "mtu":1500,
  "metric":0
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
id	1	Y	[INT]	路由列表 ID
interface	"eth0"	Y	[STRING]	网卡
target	"192.168.38.12"	Y	[STRING]	目标
netmask	"255.255.255.0"	Y	[STRING]	网络掩码
gateway	"192.168.38.254"	Y	[STRING]	网关
metric	1500	Y	[INT]	路由开销
mtu	0	Y	[INT]	最大传输单元

- 响应参数（样例）

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

- 响应参数说明

无

3.2.4 删除静态路由

- 接口描述

请求此接口可删除单个静态路由，或者批量删除部分静态路由。给定的值请参考样例

- 请求地址

/api/networkmanager/static/routing/del

- 请求方法

POST

- 请求参数（样例）

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

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
ids	1	Y	[INT]/[ARRAY(INT)]	路由 id 或 id 数组

- 响应参数（样例）

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

- 响应参数说明

无

3.2.5 静态路由开关

- 接口描述

通过请求该接口，可以开启/关闭指定路由

- 请求地址

/api/networkmanager/static/routing/set

- 请求方法

POST

- 请求参数（样例）

```
{
  "id": 1,
  "status": 1
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
id	1	Y	[INT]	路由 id
status	0/1	Y	[INT]	开启/关闭指定路由

- 响应参数（样例）

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

- 响应参数说明

无

4 Kilolink

4.1 Kilolink 客户端服务

我们的 NDI 设备提供了接入 kilolink server 的功能，可以通过 kilolink server 对设备进行集中管理。设备端通过 Kilolink 客户端服务的开启和关闭，可以对设备进行 kilolink 连接的管理。

4.1.1 获取当前设备网络接口

- 接口描述
请求本接口获取当前设备的网络接口列表，如果是普通模式将返回"eth0"和"eth1"，如果是链路聚合模式将返回"bond0"

- 请求地址
/api/kilolink/networks

- 请求方法

GET

- 请求参数（样例）
无

- 请求参数说明
无

- 响应参数（样例）

```
{  
  "result": "ok",  
  "list": [  
    "bond0"  
  ]  
}
```

- 响应参数说明

4.1.2 获取 kilolink client 设置

- 接口描述
主要提供给前端的接口，用于获取设备目前所保存的 kilolink client 设置。

- 请求地址

/api/kilolink/get

- 请求方法

GET

- 请求参数（样例）
无

- 请求参数说明
无

- 响应参数（样例）

```
{  
  "result": "ok",  
  "cfg": {  
    "ip": "",  
    "port": 50000,  
    "ifname": [],  
    "key": "",  
    "crypto": false,  
  }  
}
```

```

        "enable": false
    },
    "status": 1
}
    
```

- 响应参数说明（cfg 字段）

字段名	字段类型	字段描述
ip	[STRING]	kilolink 服务器的地址
port	[INT]	服务端口
ifname	[ARRAY(STRING)]	网络接口
key	[STRING]	授权码
crypto	[BOOLEAN]	是否加密
enable	[BOOLEAN]	是否开启 kilolink 服务

4.1.3 设置 kilolink client 配置参数

- 接口描述

通过此接口设置 kilolink client 的参数，为连接到 kilolink server 做准备

- 请求地址
/api/kilolink/set

- 请求方法

POST

- 请求参数（样例）

```

{
  "ip": "",
  "port": 50000,
  "ifname": ["eth0"],
  "key": "",
  "crypto": false,
  "enable": false,
}
    
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
ip	"192.168.41.211"	Y	[STRING]	kilolink 服务器的地址
port	5421	Y	[INT]	服务端口
ifname	["eth0"]	Y	[ARRAY(STRING)]	网络接口
key	"is8931ji"	Y	[STRING]	授权码
crypto	false	Y	[BOOLEAN]	是否加密
enable	false	Y	[BOOLEAN]	是否开启 kilolink 服务

- 响应参数（样例）

```

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

- 响应参数说明
无

4.1.4 校验 token

- 接口描述
用于校验来自于 kilolink server 的请求，验证放心

- 请求地址
`/api/kilolink/check_proxy_token?token=xxx`

- 请求方法
GET
- 请求参数（样例）
无

- 请求参数说明
无

- 响应参数（样例）

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

- 响应参数说明
无

5 语音对讲

5.1 Talk server

5.1.1 添加设备

- 接口描述
通过本接口可以给当前设备手动添加其他可进行语音对讲的设备。

- 请求地址
`/api/talk/addDevice`

- 请求方法
POST
- 请求参数（样例）

```
{
  "type": "device",
  "ip": "192.168.40.100",
  "http_port": 80,
  "https_port": 443
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
type	"device"	Y	[STRING]	类型: "device", "server"
ip	"192.168.40.100"	Y	[STRING]	设备 IP
http_port	80	Y	[INT]	http 端口号
https_port	443	Y	[INT]	https 端口号

- 响应参数（样例）

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

- 响应参数说明
无

5.1.2 设置本机设备与目标设备的连接

- 接口描述

使用本接口可以将本机与目标设备进行连接或者断开，本接口被调用后，内部逻辑会进一步对对方设备中的接口（5.1.12 的接口）发起请求。如果对方也成功进行了通话连接，即可进行语音对讲。

- 请求地址

/api/talk/setLDLinkDEnable

- 请求方法

POST

- 请求参数（样例）

```
{
  "l_dev_ip": "192.168.38.74",
  "dev_ip": "192.168.40.100",
  "enable": true
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
l_dev_ip	"192.168.38.74"	Y	[STRING]	本机设备的 IP
dev_ip	"192.168.40.100"	Y	[STRING]	目标设备的 IP
enable	true	Y	[BOOLEAN]	是否通话。 true: 建立通话连接; false:关闭通话连接

- 响应参数（样例）

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

- 响应参数说明

无

5.1.3 设置网页与目标设备的连接

- 接口描述

网页与目标设备连接，是指您可以直接在网页上，即 PC 端直接与设备进行连接，连接成功后，即可进行 PC 端与目标设备的语音通话。

- 请求地址

/api/talk/setWLinkDEnable

- 请求方法

POST

- 请求参数（样例）

```
{
  "dev_ip": "192.168.43.176",
  "enable": true
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
dev_ip	"192.168.43.176"	Y	[STRING]	目标设备的 IP
enable	true	Y	[BOOLEAN]	是否通话。 true: 建立通话连接; false:关闭通话连接

- 响应参数（样例）

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

- 响应参数说明

无

5.1.4 设置设备和语音服务器的连接

- 接口描述

通过此设备可以与语音服务器进行连接。连接成功后，即可将与当前连接在语音服务器上的所有设备进行通话。

- 请求地址

/api/talk/setLDLinkSEnable

- 请求方法

POST

- 请求参数（样例）

```
{
  "l_dev_ip": "192.168.35.11",
  "ser_ip": "192.168.40.100",
  "enable": true
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
l_dev_ip	"192.168.35.11"	Y	[STRING]	本机设备的 IP
ser_ip	"192.168.40.100"	Y	[STRING]	目标服务器的 IP
enable	true	Y	[BOOLEAN]	是否通话。 true: 建立通话连接; false:关闭通话连接

- 响应参数（样例）

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

- 响应参数说明

无

5.1.5 查询设备连接状态

- 接口描述

通过该接口可以查询当前机器的是否进行了语音连接

- 请求地址

/api/talk/queryLDeviceConnectStatus

- 请求方法

GET

- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

```
{
  "result": "ok",
  "msg": {
    "connect_status": "noconnect",
    "connect_ip": "0.0.0.0"
  }
}
```

- 响应参数说明

字段名	字段类型	字段描述
connect_status	[STRING]	设备连接状态 未连接: noconnect; 连接中: connecting; 已连接: connected
connect_ip	[STRING]	设备 ip

5.1.6 查询设备列表

- 接口描述
查询通过自动搜索和手动添加保存在设备上的所有可以进行语音对讲的设备和语音服务器。

- 请求地址
/api/talk/queryDeviceList

- 请求方法
GET

- 请求参数 (样例)
无

- 请求参数说明
无

- 响应参数 (样例)

```
{
  "result": "ok",
  "msg": {
    "local_device": { // 本机设备
      "name": "N60-248", // 设备名
      "ip": "192.168.43.248", // 设备 IP
      "type": "device", // 设备类型
      "usb_status": false, // USB 状态。true:已插入 USB, false: 未插入 USB
      "talk_status": false, // 通话状态。true: 通话中, false: 未通话
      "connect_status": "noconnect", // 连接状态。未连接: "noconnect" 连接中: "connecting" 已连接: "connected"
      "connect_ip": "0.0.0.0", // 连接对象 IP
      "connect_type": null, // 连接对象的类型。None: 没有连接 "web": 网页 "device": 设备
      "server": 服务器
      "version": null, // 本机设备通话服务的版本
      "http_port": 80, // 本地设备开放的 http 端口
      "https_port": 443 // 本地设备开放的 https 端口
    },
    "other_device": [ // 其它设备
      {
        "name": "E3-1234567891206", // 设备名
        "ip": "192.168.43.141", // 设备 IP
        "type": "device", // 设备类型。"device"或"server"
        "discovery_mode": "auto", // 发现方式。"auto":自动发现, "manual":手动添加
        "selected": false, // 是否被选中连接。true:选中本机设备与该设备进行连接, false:未选中
        "connect_status": "noconnect", // 连接状态。未连接: "noconnect" 连接中: "connecting" 已连接: "connected"
        "talk_status": "ready", // 通话状态。 "ready":设备未连接时的默认状态 "talking":设备已连接后的通话中状态 "busy":设备正在与其它对象通话
      }
    ]
  }
}
```

```

        "connect_error_info": null, // 连接失败信息
        "version": "1.10.0001", // 设备通话服务的版本
        "http_port": 80, // 设备开放的 http 端口
        "https_port": 443, // 设备开放的 https 端口
        "auth_code": null // 授权码。当设备类型为 device 时授权码为 None
    }
}
}

```

- 响应参数说明
如样例中各个字段的注释所示

5.1.7 删除设备

- 接口描述
调用此接口，传入设备的 ip 可以删除该设备。

- 请求地址
/api/talk/deleteDevice
- 请求方法
POST
- 请求参数（样例）

```

{
  "device_ip": "192.168.43.176"
}

```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
device_ip	"192.168.43.176"	Y	[STRING]	设备 IP

- 响应参数（样例）

```

{"result": "ok"}

```

- 响应参数说明
无

5.1.8 查询本机设备的序列号

- 接口描述
调用此接口，返回本机设备的序列号。

- 请求地址
/api/talk/queryLocalDeviceSerial
- 请求方法
GET
- 请求参数（样例）
无
- 请求参数说明
无
- 响应参数（样例）

```

{
  "result": "ok",
  "msg": {
    "serial": "012354640124"
  }
}

```

```
}
}
```

- 响应参数说明（msg 字段）

字段名	字段类型	字段描述
serial	[STRING]	序列号

5.1.9 talk server 版本

- 接口描述
调用此接口，可获取语音对讲服务器当前版本号

- 请求地址
/api/talk/talkVersion
- 请求方法
GET

- 请求参数（样例）
无

- 请求参数说明
无

- 响应参数（样例）

```
{
  "result": "ok",
  "msg": {
    "version": "1.10.0001"
  }
}
```

- 响应参数说明

字段名	字段类型	字段描述
version	[STRING]	版本字段

5.1.10 设置语言

- 接口描述
调用此接口，可以设置语音对讲模块在后台网页上的显示语言

- 请求地址
/api/talk/setLanguage

- 请求方法
POST
- 请求参数（样例）

```
{
  "language": "cn"
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
language	"cn"	Y	[STRING]	cn: 英文模式; zh: 中文模式

- 响应参数（样例）

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

- 响应参数说明
无

6 音视频录制

6.1 存储管理

6.1.1 添加 nas 网盘

- 接口描述

本接口主要是用于为设备添加需要被挂载的 nas 网盘, 从而让设备能够将录制的文件存放在指定的 nas 网盘之上。

- 请求地址
api/record/storage/add_nas
- 请求方法
POST
- 请求参数 (样例)

```
{
  "id": "public",
  "name": "public",
  "type": "nfs",
  "host": "192.168.0.21",
  "volume": "/mnt/disk-data/public",
  "options": ""
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
id	"public"	Y	[STRING]	需要连接的 nas 网盘的 id, 自定义即可。
name	"public"	Y	[STRING]	需要连接的 nas 网盘的名称, 自定义即可。
type	"nfs"	Y	[STRING]	选择需要连接到的 nas 的文件系统类型
host	"192.168.0.21"	Y	[STRING]	远程 nas 的主机 ip
volume	"/mnt/disk-data/public"	Y	[STRING]	那是上需要挂载到本机的目录
options	""	Y	[STRING]	其他操作, 比如 cifs 文件系统的 nas 的用户名和密码就在这里设置。

- 响应参数 (样例)

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

- 响应参数说明
无

6.1.2 移除 nas 网盘

- 接口描述

该接口主要移除指定的 nas 网盘

- 请求地址
/api/record/storage/remove_nas
- 请求方法
GET
- 请求参数（样例）

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

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
id	"public"	Y	[STRING]	需要删除的 nas 网盘的 id，自定义即可。

- 响应参数（样例）

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

- 响应参数说明

无

6.1.3 获取已挂载磁盘列表

- 接口描述

本接口主要是获取当前系统种已经挂载的磁盘列表，包括已经挂载上的 U 盘，以及挂载上或者正在挂载种的 nas 盘

- 请求地址
/api/record/storage/disk_list
- 请求方法
GET
- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "nas": [],
    "usb": [
      {
        "id": "sda1",
        "name": "sda1",
        "type": null,
        "host": null,
        "volume": "/dev/sda1",
        "options": null,
        "mounted": "/run/media/sda1",
        "filesys": "vfat",
        "status": "mounted",
        "choose": true,
        "disk_type": "usb",
        "total": 29494,
      }
    ]
  }
}
```

```

        "used": 9139,
        "free": 20354,
        "percent": 31
    },
    {
        "id": "mmcblk0p1",
        "name": "mmcblk0p1",
        "type": null,
        "host": null,
        "volume": "/dev/mmcblk0p1",
        "options": null,
        "mounted": "/media/sd-mmcblk0p1",
        "filesystem": "vfat",
        "status": "mounted",
        "choose": true,
        "disk_type": "usb",
        "total": 499,
        "used": 141,
        "free": 359,
        "percent": 28
    }
]
}

```

● 响应参数说明（data 字段，以 usb 盘为例）

字段名	字段类型	字段描述
id	[STRING]	设备自定义的连接 id 号，nas 盘的话自己随意输入，usb 由系统生成
name	[STRING]	设备自定义的连接名称，nas 盘的话自己随意输入，usb 由系统生成
type	[STRING]	连接类型
host	[STRING]	如果是 usb 盘，该值为 null；如果是 nas 盘，该值为 nas 主机的地址。
volume	[STRING]	usb 盘或者 nas 盘设备的地址
options	[STRING]	如果是 nas 盘的话，就是
mounted	[STRING]	磁盘在设备上用于保存录像的挂载目录
filesystem	[STRING]	文件系统类型
status	[STRING]	当前磁盘的挂载状态
choose	[BOOLEAN]	是否被选中作为需要录制的磁盘
disk_type	[STRING]	磁盘类型
total	[INT]	磁盘空间总量（MB）
used	[INT]	磁盘已经使用的空间大小（MB）
free	[INT]	磁盘剩余空间大小（MB）
percent	[INT]	磁盘已经使用的空间的百分比

6.1.4 获取录制的文件列表

- 接口描述
本接口用于获取已录制和正在录制的文件列表，因目前只开放了 NDI|HX 的录制，所以获取到的文件均来自 NDI|HX 编码。

- 请求地址
api/record/storage/get_disk_files?device=/run/media/sda1
 or
api/record/storage/get_disk_files?device=/run/media/sda1&stream=ndi_hx

- 请求方法
GET
- 请求参数（样例）
如请求地址所示

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
device	/run/media/sda1	Y	[STRING]	设备在本机的挂载地址
stream	ndi_hx	N	[STRING]	需要查询的录制的编码类型

- 响应参数（样例）

```
{
  "result": "ok",
  "data": [
    {
      "path":
"/run/media/sda1/RECORD/M20230922_172235_00232/REC-NDI_HX-20230922172235002320.mov",
      "name": "REC-NDI_HX-20230922172235002320.mov",
      "size": 66.35,
      "current": false,
      "stream": "ndi_hx",
      "start_time": "2023-09-22 17:23:30",
      "end_time": "2023-09-22 17:24:28"
    },
    {
      "path":
"/run/media/sda1/RECORD/M20230922_170223_747971/REC-NDI_HX-202309221702237479710.mov",
      "name": "REC-NDI_HX-202309221702237479710.mov",
      "size": 33.79,
      "current": false,
      "stream": "ndi_hx",
      "start_time": "2023-09-22 17:02:51",
      "end_time": "2023-09-22 17:03:28"
    },
    {
      "path":
"/run/media/sda1/RECORD/M20230922_165741_746751/REC-NDI_HX-202309221657417467511.mov",
      "name": "REC-NDI_HX-202309221657417467511.mov",
      "size": 24.53,
      "current": false,
      "stream": "ndi_hx",
      "start_time": "2023-09-22 16:58:01",
      "end_time": "2023-09-22 16:59:02"
    }
  ]
}
```

```

    },
    {
      "path":
"/run/media/sda1/RECORD/M20230922_165622_761459/REC-NDI_HX-202309221656227614590.mov",
      "name": "REC-NDI_HX-202309221656227614590.mov",
      "size": 96.6,
      "current": false,
      "stream": "ndi_hx",
      "start_time": "2023-09-22 16:57:41",
      "end_time": "2023-09-22 16:59:00"
    }
  ]
}

```

- 响应参数说明（data 字段，以 usb 盘为例）

字段名	字段类型	字段描述
path	[STRING]	录制文件的存放路径
name	[STRING]	录制文件的文件名
size	[STRING]	录制文件的大小
current	[BOOLEAN]	当前文件是否正在录制
stream	[STRING]	流的名称
start_time	[STRING]	录制开始时间
end_time	[STRING]	录制结束时间

6.1.5 删除录制文件

- 接口描述
该接口用于删除指定的录制文件

- 请求地址
/api/record/storage/remove_file
- 请求方法
POST
- 请求参数（样例）

```
{"path": "/run/media/sda1/RECORD/M20230922_165622_761459/REC-NDI_HX-202309221656227614590.mov"}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
path	"/run/media/sda1/RECORD/M20230922_165622_761459/REC-NDI_HX-202309221656227614590.mov"	Y	[STRING]	需要删除的录制文件的文件路径

- 响应参数（样例）

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

- 响应参数说明

无

6.2 录制配置管理

6.2.1 录制信息设置

- 接口描述
该接口主要用于设置录制信息，包括录制中的分片时间、大小，录制模式，选中需要录制的磁盘等。

- 请求地址
/api/record/record/settings
- 请求方法
POST
- 请求参数（样例）

```
{
  "disk_choose": {"nas": ["public"], "usb": ["sda1"]} # 勾选上的存储盘子在这个位置记录下选择的 nas 或者 usb 盘的 id
  "auto_record": True, # 自动录制，当磁盘就绪时，如果录制功能开启，自动写入磁盘
  "format": "mov", # 录像文件格式
  "limit_type": "size-loop", # 文件限制，None/size-loop/time-loop/size/time
  "limit_size": 100 * 1024, # 限制大小（Mb）
  "limit_time": 60 * 60, # 限制时长（s）
  "disk_policy": "overwrite", # 存储策略，如：覆盖旧录像 overwrite/noSpace
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
disk_choose	{"nas": ["public"], "usb": ["sda1"]}	Y	[OBJECT]	勾选上的存储盘子在这个位置记录下选择的 nas 或者 usb 盘的 id
auto_record	True	N	[BOOLEAN]	自动录制，当磁盘就绪时，如果录制功能开启，自动写入磁盘。（注意：此功能暂时关闭，参数仅作参考）
format	"mov"	Y	[STRING]	录像文件的封装格式
limit_type	"size-loop"	Y	[STRING]	文件限制，None/size-loop/time-loop/size/time
limit_size	100 * 1024	Y	[INT]	限制大小（Mb）
limit_time	60 * 60	Y	[INT]	限制时长（s）
disk_policy	"overwrite"	Y	[STRING]	存储策略，如：覆盖旧录像 overwrite/noSpace

- 响应参数（样例）
无
- 响应参数说明
无

6.2.2 获取录制信息

- 接口描述
用于显示已经设置好并保存的录制信息。

- 请求地址
/api/record/record/get_record_info
- 请求方法
GET
- 请求参数（样例）

无

- 请求参数说明

无

- 响应参数（样例）

```
{
  "result": "ok",
  "data": {
    "auto_record": true, # 是否自动录制
    "format": "mov", # 录制的视频格式
    "limit_type": "size-loop", # 文件限制, None/size-loop/time-loop/size/time
    "limit_size": 4000000, # 限制大小 (Mb)
    "limit_time": 3600, # 限制时长 (s)
    "disk_policy": "overwrite" # 存储策略, 如: 覆盖旧录像/录制到满 overwrite/noSpace
  }
}
```

- 响应参数说明（data 字段，以 usb 盘为例）

字段名	字段类型	字段描述
auto_record	[STRING]	是否自动录制。（注意：此功能暂时关闭，参数仅作参考）
format	[STRING]	录制的视频格式
limit_type	[STRING]	限制类型： None：不限 size-loop：限制大小，自动分割多个文件 time-loop：限制时长，自动分割多个文件
limit_size	[BOOLEAN]	限制大小（Mb）
limit_time	[STRING]	限制时长（s）
disk_policy	[STRING]	存储策略，如： overwrite：覆盖旧录像 noSpace：录制到满

6.3 录制

6.3.1 录制启停

- 接口描述

该接口主要用于手动启动和停止录制

- 请求地址

/api/record/ndi_hx/recording

- 请求方法

POST

- 请求参数（样例）

```
{
  "start": true
}
```

- 请求参数说明

参数名称	参数值样例	是否必填	参数类型	参数描述
------	-------	------	------	------

start	true	Y	[BOOLEAN]	true: 开启录制; false: 关闭录制
-------	------	---	-----------	----------------------------

- 响应参数 (样例)
无
- 响应参数说明
无