

# Pro Convert AES67

## 前言

<a href="#">文档简介</a>	1.1
<a href="#">接口约定</a>	1.2
<a href="#">API 状态码</a>	1.3
<a href="#">DEMO: 命令行</a>	1.4

## 通用接口

<a href="#">ping</a>	2.1
<a href="#">reboot</a>	2.2
<a href="#">factory-reset-permission</a>	2.3
<a href="#">factory-reset</a>	2.4

## 设备状态

<a href="#">summary-info</a>	3.1
------------------------------	-----

## 全局设置

<a href="#">general-info</a>	4.1
<a href="#">general-settings</a>	4.2

## PTP

<a href="#">ptp-info</a>	5.1
<a href="#">ptp-settings</a>	5.2
<a href="#">ptp-default</a>	5.3
<a href="#">ptp-state</a>	5.4

## AES67 发送

<a href="#">source-info</a>	6.1
<a href="#">source-apply</a>	6.2
<a href="#">source-del</a>	6.3

## AES67 接收

<a href="#">sink-info</a>	7.1
<a href="#">sink-apply</a>	7.2
<a href="#">sink-del</a>	7.3
<a href="#">sink-statistics</a>	7.4
<a href="#">sink-statistics-clear</a>	7.5
<a href="#">sdp-parse</a>	7.6

## Stream 通用

<a href="#">ices-receiver-select</a>	8.1
--------------------------------------	-----

## Stream NDI TX

<a href="#">ndi-sender-info</a>	9.1
<a href="#">ndi-sender-apply</a>	9.2
<a href="#">ndi-sender-del</a>	9.3
<a href="#">ndi-sender-find</a>	9.4
<a href="#">ndi-sender-options</a>	9.5

## Stream NDI RX

<a href="#">ndi-receiver-info</a>	10.1
<a href="#">ndi-receiver-apply</a>	10.2
<a href="#">ndi-receiver-del</a>	10.3
<a href="#">ndi-receiver-find</a>	10.4
<a href="#">ndi-receiver-options</a>	10.5

## Stream SRT TX

<a href="#">srt-sender-info</a>	11.1
<a href="#">srt-sender-apply</a>	11.2
<a href="#">srt-sender-del</a>	11.3

## Stream SRT RX

<a href="#">srt-receiver-info</a>	12.1
<a href="#">srt-receiver-apply</a>	12.2
<a href="#">srt-receiver-del</a>	12.3

## Stream ICES TX

<a href="#">ices-sender-info</a>	13.1
<a href="#">ices-sender-apply</a>	13.2
<a href="#">ices-sender-del</a>	13.3

## Stream ICES RX

<a href="#">ices-receiver-info</a>	14.1
<a href="#">ices-receiver-apply</a>	14.2
<a href="#">ices-receiver-del</a>	14.3
<a href="#">ices-list-streams</a>	14.4

## 矩阵

<a href="#">matrix-info</a>	15.1
<a href="#">audio-meter</a>	15.2
<a href="#">audio-meter-limit</a>	15.3
<a href="#">matrix-meter</a>	15.4
<a href="#">matrix-settings</a>	15.5
<a href="#">matrix-clear</a>	15.6
<a href="#">volume-tx</a>	15.7
<a href="#">volume-mix</a>	15.8

## 系统设置

device-info	16.1
info	16.2
set-device-name	16.3
set-date-time	16.4
timezone-set	16.5

## 网络设置

if-info	17.1
if-set	17.2
if-route	17.3
get-dns	17.4
set-dns	17.5
usb-config	17.6

## 用户

login	18.1
logout	18.2
get-all	18.3
add	18.4
del	18.5
ch-password	18.6
set-password	18.7

## 固件更新

online-check	19.1
online-check-result	19.2
upload-fw	19.3
update	19.4
state	19.5
clear	19.6

## 系统日志

clear	20.1
filter	20.2
export	20.3

## 文档简介

针对AES67，我们开放了丰富的 API，方便开发人员与设备交互，如获取设备的基本信息（设备名称、固件版本等），修改设备配置，更新固件等。这些 API 基于 HTTP 协议，是一种轻量级、无连接状态的接口，响应数据为 JSON 格式。通过本文档，您可以更详细地了解每个 API 的功能和请求方式。

本文档中的 API 可适用于以下产品：

- AES67

# 接口约定

## 一、概要

- 请求协议: HTTP
- 请求方式: 默认情况下, 数据请求和提交都用 POST(application/json)
- 返回数据格式: HTTP 状态为 200 时, 返回 JSON 数据, 否则为 HTTP 对应错误
- 登录认证方式: 在 Cookie 中携带 sid=xxxxxxx
- 文件上传: 访问方式: POST(multipart/form-data)

## 二、返回 JSON 数据格式

格式如下, JSON 对象中的 status 属性为 [API 状态码](#), 为 0 时表示数据获取或操作成功, 否则为相应的失败状态码。

```
{
  status: 0,
  enable: true,
  enable-web-control: true
  ...
}
```

## API 状态码

```
{
  0: MW_STATUS_SUCCESS,
  1: MW_STATUS_PENDING,
  2: MW_STATUS_TIMEOUT,
  3: MW_STATUS_INTERRUPTED,
  4: MW_STATUS_TRY_AGAIN,
  5: MW_STATUS_NOT_IMPLEMENTED,
  6: MW_STATUS_UNKNOWN_ERROR,
  7: MW_STATUS_INVALID_ARG,
  8: MW_STATUS_NO_MEMORY,
  9: MW_STATUS_UNSUPPORTED,
  10: MW_STATUS_FILE_BUSY,
  11: MW_STATUS_DEVICE_BUSY,
  12: MW_STATUS_DEVICE_LOST,
  13: MW_STATUS_IO_FAILED,
  14: MW_STATUS_READ_FAILED,
  15: MW_STATUS_WRITE_FAILED,
  16: MW_STATUS_NOT_EXIST,
  17: MW_STATUS_TOO_MANY,
  18: MW_STATUS_TOO_LARGE,
  19: MW_STATUS_OVERFLOW,
  20: MW_STATUS_UNDERFLOW,
  21: MW_STATUS_FORMAT_ERROR,
  22: MW_STATUS_FILE_EXISTS,
  23: MW_STATUS_FILE_TYPE_ERROR,
  24: MW_STATUS_DEVICE_TYPE_ERROR,
  25: MW_STATUS_IS_DIRECTORY,
  26: MW_STATUS_READ_ONLY,
  27: MW_STATUS_RANGE_ERROR,
  28: MW_STATUS_BROKEN_PIPE,
  29: MW_STATUS_NO_SPACE,
  30: MW_STATUS_NOT_DIRECTORY,
  31: MW_STATUS_NOT_PERMITTED,
  32: MW_STATUS_BAD_ADDRESS,
  33: MW_STATUS_SEEK_ERROR,
  34: MW_STATUS_CROSS_DEVICE_LINK,
  35: MW_STATUS_NOT_INITIALIZED,
  36: MW_STATUS_AUTH_FAILED,
  37: MW_STATUS_NOT_LOGGED_IN,
  38: MW_STATUS_WRONG_STATE,
  39: MW_STATUS_MISMATCH,
  40: MW_STATUS_VERIFY_FAILED,
  41: MW_STATUS_CONSTRAINT_VIOLATION,
  42: MW_STATUS_CANCELED,
  43: MW_STATUS_IN_PROGRESS,
  44: MW_STATUS_CONN_REFUSED,
  45: MW_STATUS_CONN_RESET,
  46: MW_STATUS_ADDR_IN_USE,
  47: MW_STATUS_NO_RESPONSE,
  48: MW_STATUS_INFO_CHANGED,
  49: MW_STATUS_INVALID_DATA,
  50: MW_STATUS_NEED_MORE_DATA,
  51: MW_STATUS_NO_BUFFER,
  52: MW_STATUS_BUFFER_TOO_SMALL,
  53: MW_STATUS_BUFFER_IS_EMPTY,
  54: MW_STATUS_BUFFER_IS_FULL
}
```

## DEMO: 命令行

在不同操作系统中，可以安装 wget 和 curl 两个工具，安装后可以在命令行中通过 wget 或 curl 命令来调用 API。

不同操作系统中，下边示例的 cookie 文件存放位置不同，请根据实际情况修改。(以下示例基于Linux。)

### wget

#### 1.登录并保存 cookies

```
wget --save-cookies=sid.txt --keep-session-cookies --header="Content-Type: application/json" --post-data='{"username":"Admin", "password": "c1c224b03cd9bc7b6a86d77f5dace40191766c485cd55dc48caf9ac873335d6f"}' http://192.168.66.1/api/user/login -d -q -O -
```

#### 2.获取用户列表

```
wget --load-cookies=sid.txt --keep-session-cookies --header="Content-Type: application/json" --post-data='' http://192.168.66.1/api/user/get-all -d -q -O -
```

#### 3.新增用户

```
wget --load-cookies=sid.txt --keep-session-cookies --header="Content-Type: application/json" --post-data='{"username":"test", "password": "9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08"}' http://192.168.66.1/api/user/add -d -q -O -
```

### curl

#### 1.登录并保存 cookies

```
curl --cookie-jar sid.txt http://192.168.66.1/api/user/login -X POST -H 'Content-Type: application/json' -d '{"username":"Admin", "password": "c1c224b03cd9bc7b6a86d77f5dace40191766c485cd55dc48caf9ac873335d6f"}'
```

#### 2.获取用户列表

```
curl --cookie sid.txt http://192.168.66.1/api/user/get-all -X POST -H 'Content-Type: application/json' -d ''
```

#### 3.新增用户

```
curl --cookie sid.txt http://192.168.66.1/api/user/add -X POST -H 'Content-Type: application/json' -d '{"username":"test", "password": "9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08"}'
```

## ping 接口

判断设备是否可以访问，无需登录。

在 [固件更新](#)、[重置设备](#)、[修改 IP 地址](#) 等操作完成后，设备需要重启，可以通过该接口判断设备是否已经重启完成。

### 请求方式

```
GET/POST /api/ping
```

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。



## reboot 接口

此接口用于重启设备，重启后需要重新登录，仅管理员有权限。

重启过程需要几分钟时间，可以使用 [ping 接口](#) 判断设备是否已经重启。

### 请求方式

GET/POST /api/reboot

### 返回数据

```
{
  "status": 0,
  "delay": 5,
  "estimate-sec": 15
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
delay	延时多少秒执行重启，单位 s。
estimate-sec	估算重启时间，单位 s。

# factory-reset-permission 接口

通过该接口判断是否允许重置设备, 无需登录。

## 请求方式

GET/POST /api/factory-reset-permission

## 返回数据

```
{  
  "status": 0  
  "reset-enable": true  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
reset-enable	是否支持重置功能, 有效值: true/false。

## factory-reset 接口

恢复出厂设置。

### 请求方式

GET/POST /api/factory-reset

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /api/aes67/summary-info 接口

获取 AES67 运行信息。

### 请求方式

POST /api/aes67/summary-info

### 返回数据

```
{
  "status": 0,
  "temperature": "79.61°C",
  "card-address": 0,
  "unbalance": {
    "in": {
      "linked": true,
      "depth": "L24",
      "channel-num": 2
    },
    "out": {
      "linked": false,
      "depth": "L24",
      "channel-num": 2
    }
  },
  "balance": {
    "in": {
      "linked": false,
      "depth": "L24",
      "channel-num": 2
    },
    "out": {
      "linked": false,
      "depth": "L24",
      "channel-num": 2
    }
  },
  "uac": {
    "usb-connected": true,
    "in": {
      "linked": false,
      "sample-rate": 48000,
      "depth": "L24",
      "channel-num": 4
    },
    "out": {
      "linked": false,
      "sample-rate": 48000,
      "depth": "L24",
      "channel-num": 4
    }
  },
  "sources": [
    {
      "name": "Pro_Convert_AES67_7005_0_01",
      "depth": "L24",
      "channel-num": 2,
      "sample-rate": 48000,
      "ptime": "1ms"
    }
  ],
  "sinks": [
    {
      "name": "Pro_Convert_AES67_1015_0_01",
      "depth": "L24",
      "channel-num": 2,

```

```

    "sample-rate": "48000",
    "ptime": "1ms",
    "qos": 2
  }
]
}

```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
temperature	温度。
card-address	拨码开关地址。
unbalance.in.linked	unbalance输入是否连接, true: 连接, false: 未连接
unbalance.in.depth	unbalance输入采样位宽 L16, L24
unbalance.in.channel-num	unbalance输入通道数
unbalance.out.linked	unbalance输出是否连接
unbalance.out.depth	unbalance输出采样位宽 L16, L24
unbalance.out.channel-num	unbalance输出通道数
balance.in.linked	balance输入是否连接
balance.in.depth	balance输入采样位宽 L16, L24
balance.in.channel-num	balance输入通道数
balance.out.linked	balance输出是否连接
balance.out.depth	balance输出采样位宽 L16, L24
balance.out.channel-num	balance输出通道数
uac.usb-connected	USB 是否已连接, true: 连接, false: 未连接
uac.in.linked	UAC输入是否工作 true/false
uac.in.sample-rate	UAC输入采样率
uac.in.depth	UAC 输入采样位宽
uac.in.channel-num	UAC输入通道数
uac.out.linked	UAC输出是否工作 true/false
uac.out.sample-rate	UAC输出采样率
uac.out.depth	UAC输出采样位宽
uac.out.channel-num	UAC输出通道数
sources[i].name	AES67 source名称
sources[i].sample-rate	AES67 source 采样率, 支持44100, 48000, 96000
sources[i].depth	AES67 source采样位宽 L16, L24
sources[i].channel-num	AES67 source 通道数
sources[i].ptime	AES67 packet time, 支持0.125ms, 0.250ms, 0.333ms, 1ms, 4ms
sinks[i].name	AES67 sink 连接的source 名称
sinks[i].sample-rate	AES67 sink 采样率, 支持44100, 48000, 96000
sinks[i].depth	AES67 sink 采样位宽 L16, L24
sinks[i].channel-num	AES67 sink通道数
sinks[i].ptime	AES67 packet time, 支持0.125ms, 0.250ms, 0.333ms, 1ms, 4ms
sinks[i].qos	Qos值的组合 0x0001: sync 0x0002: not sync 0x0004: Underflow 0x0008: Overflow 0x0010: Late 0x0020: unmute 0x0040: mute



# /api/aes67/general-info 接口

获取AES67 通用设置信息

## 请求方式

POST /api/aes67/general-info

## 返回数据

```
{
  "status": 0,
  "tx-sample-rate": 48000,
  "tx-ptime": "1ms",
  "unbld-in": 2,
  "unbld-out": 2,
  "bld-in": 1,
  "bld-out": 0,
  "micbias": true,
  "st2110": false,
  "igmp": 0,
  "aac-bitrate-kbps": 128,
  "uac-num-channels": 4
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
tx-sample-rate	AES67 source采样率, 支持44100, 48000, 96000
tx-ptime	AES67 source的packet time, 支持0.125ms, 0.250ms, 0.333ms, 1ms, 4ms
unbld-in	0: +12dBu(default) 1: +4dBu(SMPTE) 2: +0dBu 3: -2dBu(EBU) 4: 0dBV 5: -10dBV
unbld-out	0: +12dBu(default) 1: +4dBu(SMPTE) 2: +0dBu 3: -2dBu(EBU) 4: 0dBV 5: -10dBV
bld-in	0: +24dBu 1: +18dBu(default) 2: +4dBu(SMPTE) 3: +0dBu 4: -2dBu(EBU) 5: 0dBV 6: -10dBV
bld-out	0: +18dBu(default) 1: +4dBu(SMPTE) 2: +0dBu 3: -2dBu(EBU) 4: 0dBV 5: -10dBV
micbias	MIC 供电电压是否使能true/false
igmp	IGMP 版本, 0:自动 2: IGMPv2 3:IGMPv3
aac-bitrate-kbps	Stream TX AAC码率, 支持128,192,256
uac-num-channels	UAC 通道数, 设置使用/system/uac-format

# /api/aes67/general-settings 接口

AES67 通用设置

## 请求方式

POST /api/aes67/general-settings

属性	说明
tx-sample-rate	source采样率, 支持44100, 48000, 96000
tx-ptime	source的packet time, 支持0.125ms, 0.250ms, 0.333ms, 1ms, 4ms
unbld-in	0: +12dBu(default) 1: +4dBu(SMPTE) 2: +0dBu 3: -2dBu(EBU) 4: 0dBV 5: -10dBV
unbld-out	0: +12dBu(default) 1: +4dBu(SMPTE) 2: +0dBu 3: -2dBu(EBU) 4: 0dBV 5: -10dBV
bld-in	0: +24dBu 1: +18dBu(default) 2: +4dBu(SMPTE) 3: +0dBu 4: -2dBu(EBU) 5: 0dBV 6: -10dBV
bld-out	0: +18dBu(default) 1: +4dBu(SMPTE) 2: +0dBu 3: -2dBu(EBU) 4: 0dBV 5: -10dBV
micbias	MIC 供电电压是否使能true/false
igmp	IGMP 版本, 0:自动 2: IGMPv2 3:IGMPv3
aac-bitrate-kbps	Stream TX AAC码率, 支持128,192,256
uac-num-channels	UAC 通道数, 设置使用/system/uac-format

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。



# /api/aes67/ptp-info 接口

PTP 服务运行信息

## 请求方式

POST /api/aes67/ptp-info

## 返回数据

```
{
  "status": 0,
  "version": "PTPV2",
  "priority1": 255,
  "priority2": 255,
  "domain": 0,
  "gmid": "D0-C8-57-FF-FE-80-BD-5A",
  "role": "Slaver",
  "delay-mech": "E2E",
  "announce": -2,
  "sync": -1
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
version	PTP协议版本, 可用值: PTPV1, PTPV2
priority1	优先级1
priority2	优先级2
domain	PTP domain, 范围[0,127]
gmid	GMID信息
role	PTP master/slaver 选择, 可用值: Auto, Slaver
delay-mech	延时机制, 可用值: E2E, P2P
announce	可用值[-3,1]
sync	可用值[-3,1]

# /api/aes67/ptp-settings 接口

设置PTP服务

## 请求方式

POST /api/aes67/ptp-settings

属性	说明
version	PTP协议版本, 可用值: PTPV1, PTPV2
priority1	优先级1
priority2	优先级2
domain	PTP domain, 范围[0,127]
role	PTP master/slaver 选择, 可用值: Auto, Slaver
delay-mech	延时机制, 可用值: E2E, P2P
announce	可用值[-3,1]
sync	可用值[-3,1]

## 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/aes67/ptp-default 接口

PTP 服务恢复默认参数

## 请求方式

POST /api/aes67/ptp-default

## 返回数据

```
{  
  "status": 0,  
  "version": "PTPV2",  
  "priority1": 255,  
  "priority2": 255,  
  "domain": 0,  
  "role": "Slaver",  
  "delay-mech": "E2E",  
  "announce": -2,  
  "sync": -1  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
version	PTP协议版本, 可用值: PTPV1, PTPV2
priority1	优先级1
priority2	优先级2
domain	PTP domain, 范围[0,127]
role	PTP master/slaver 选择, 可用值: Auto, Slaver
delay-mech	延时机制, 可用值: E2E, P2P
announce	可用值[-3,1]
sync	可用值[-3,1]

# /api/aes67/ptp-state 接口

PTP 服务运行状态

## 请求方式

POST /api/aes67/ptp-state

## 返回数据

```
{
  "status": 0,
  "is-sync": true,
  "is-master": false,
  "lock-to": "D0-C8-57-FF-FE-80-DF-9D",
  "statistics": {
    "datetime": "2022-10-14 14:26:03",
    "interval": 2000,
    "delay": [
      -81,
      286,
      -182,
      -70,
      -84,
      -143,
      -393,
      -35,
      516,
      6,
      -772,
      -382,
      -269,
      -83,
      -23
    ]
  }
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
is-sync	PTP是否同步
is-master	true: master, false: slaver
lock-to	PTP锁定的GMID
statistics.datetime	设备当前时间
statistics.interval	统计间隔, 单位ms
statistics.delay	延时记录, 单位ns

# /api/aes67/source-info 接口

AES67 source 信息

## 请求方式

POST /api/aes67/source-info

## 返回数据

```
{
  "status": 0,
  "sources": [
    {
      "uid": 7734,
      "name": "Pro_Convert_AES67_7005_0_01",
      "enable": true,
      "state": 1,
      "dst-ip": "239.129.83.91",
      "depth": "L24",
      "channel-num": 2,
      "ttl": 15,
      "dscp": "EF",
      "sdp": "v=0\r\no=- 0 200 IN IP4 10.10.10.202\r\ns=Pro_Convert_AES67_7005_0_01\r\nc=IN IP4 239.129.83.91/15\r\nt=0 0\r\na=keywds:Magewell\r\nm=audio 5004 RTP/AVP 97\r\ni=2 channels: Left, Right\r\na=recvonly\r\na=rtpmap:97 L24/48000/2\r\na=ptime:1\r\na=ts-refclk:ptp=IEEE1588-2008:D0-C8-57-FF-FE-80-BD-5A:0\r\na=mediaclock:direct=0\r\na=ssrc:0\r\n"
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
sources[i].uid	唯一UID
sources[i].name	AES67 source名称
sources[i].enable	是否使能
sources[i].state	发送状态 0: pending 1: sending 2: fail 3:单播地址不可用
sources[i].dst-ip	source发送IP
sources[i].depth	采样位宽 L16, L24
sources[i].channel-num	通道数
sources[i].ttl	TTL值, 范围[0, 255]
sources[i].dscp	DSCP可用值 BE, AF31, AF41, EF
sources[i].sdp	SDP 信息

# /api/aes67/source-apply 接口

AES67应用source 配置

## 请求方式

POST /api/aes67/source-apply

属性	说明
uid	唯一UID
name	AES67 source名称
enable	是否使能
dst-ip	source发送IP
depth	采样位宽 L16, L24
channel-num	通道数
ttl	TTL值, 范围[0, 255]
dscp	DSCP可用值 BE, AF31, AF41, EF

## 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /api/aes67/source-del 接口

AES67删除source 配置

### 请求方式

POST /api/aes67/source-del

属性	说明
uid	唯一UID

### 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/aes67/sink-info 接口

AES67 sink信息

## 请求方式

POST /api/aes67/sink-info

## 返回数据

```
{
  "status": 0,
  "sinks": [
    {
      "sink-idx": 0,
      "state": 2,
      "name": "Pro_Convert_AES67_1015_0_01",
      "is-manual": false,
      "source-name": "Pro_Convert_AES67_1015_0_01",
      "source-src-ip": "10.10.4.195",
      "source-src-mac": "d0:c8:57:80:df:9d",
      "source-dst-ip": "239.178.203.32",
      "source-dst-port": 5004,
      "ssrc": 425112524,
      "depth": "L24",
      "channel-num": 2,
      "sample-rate": "48000",
      "ptime": "1ms",
      "mediaclk-offset": 0,
      "max-delay": 0,
      "max-buffer": 6,
      "ttl": 15,
      "sdp": ""
    }
  ],
  "sources": [
    {
      "source-name": "Pro_Convert_AES67_1015_0_01",
      "source-src-ip": "10.10.4.195",
      "source-src-mac": "d0:c8:57:80:df:9d",
      "source-dst-ip": "239.178.203.32",
      "source-dst-port": 5004,
      "ssrc": 425112524,
      "depth": "L24",
      "channel-num": 2,
      "sample-rate": "48000",
      "ptime": "1ms",
      "mediaclk-offset": 0,
      "ttl": 15,
      "sdp": ""
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
sinks[i].sink-idx	sink组编号, 范围[0,3]
sinks[i].state	AES67 sink 工作状态 0:idle 1:success 2:connecting 3:lost 4:format error
sinks[i].name	名称
sinks[i].is-manual	手动添加



sinks[i].source-name	source名称
sinks[i].source-src-ip	source 源IP信息
sinks[i].source-src-mac	source 源Mac信息
sinks[i].source-dst-ip	source 目标地址
sinks[i].source-dst-port	source 目标端口
sinks[i].ssrc	source RTP SSRC
sinks[i].sample-rate	source采样率
sinks[i].depth	采样位宽 L16, L24
sinks[i].channel-num	设备通道数
sinks[i].ptime	packet time, 支持0.125ms, 0.250ms, 0.333ms, 1ms, 4ms
sinks[i].mediaclk-offset	RTP offset信息
sinks[i].max-delay	最大延时, 范围[0, 50], 单位ms, 0表示不关心延时
sinks[i].max-buffer	最大多少个ptime, 范围[1, 16]
sinks[i].ttl	TTL值, 范围[1, 255]
sinks[i].sdp	SDP 信息
sources[i].source-src-ip	source 源IP信息
sources[i].source-src-mac	source 源Mac信息
sources[i].source-dst-ip	source 目标地址
sources[i].source-dst-port	source 目标端口
sources[i].ssrc	source RTP SSRC
sources[i].sample-rate	source 采样率, 设备目前仅支持44100, 48000, 96000
sources[i].ptime	packet time, 支持0.125ms, 0.250ms, 0.333ms, 1ms, 4ms
sources[i].depth	采样率 L16, L24
sources[i].channel-num	设备通道数
sources[i].mediaclk-offset	RTP offset信息
sources[i].ttl	TTL值, 范围[1, 255]
sources[[i].sdp	SDP 信息

# /api/aes67/sink-apply 接口

AES67设置sink

## 请求方式

POST /api/aes67/sink-apply

属性	说明
sinks[i].sink-idx	sink组编号, 范围[0,3]
sinks[i].name	名称
sinks[i].is-manual	手动添加
sinks[i].source-name	source名称
sinks[i].source-src-ip	source 源IP信息
sinks[i].source-src-mac	source 源Mac信息
sinks[i].source-dst-ip	source 目标地址
sinks[i].source-dst-port	source 目标端口
sinks[i].ssrc	source RTP SSRC
sinks[i].sample-rate	设置采样率, 设备目前仅支持44100, 48000, 96000
sinks[i].depth	采样位宽 L16, L24
sinks[i].channel-num	设备通道数
sinks[i].ptime	packet time, 支持0.125ms, 0.250ms, 0.333ms, 1ms, 4ms
sinks[i].mediaclock-offset	RTP offset信息
sinks[i].max-delay	最大延时, 范围[0, 50], 单位ms, 0表示不关心延时
sinks[i].max-buffer	最大多少个ptime, 范围[1, 16]

## 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/aes67/sink-del 接口

AES67删除sink

## 请求方式

POST /api/aes67/sink-del

属性	说明
sinks[i].sink-idx	sink组编号, 范围[0,3]

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/aes67/sink-statistics 接口

AES67 sink信息

## 请求方式

POST /api/aes67/sink-statistics

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
sink-idx	sink组编号, 范围[0,3]

## 返回数据

```
{
  "status": 0,
  "qos": 2,
  "duration": 4690,
  "peak": 0,
  "average": 0,
  "late": 0,
  "lost": 0,
  "buff-count": 0,
  "buff-underflow": 0,
  "buff-overflow": 0,
  "statistics": [
    {
      "x-us": 5000,
      "y-count": 0
    },
    {
      "x-us": 10000,
      "y-count": 0
    },
    {
      "x-us": 15000,
      "y-count": 0
    },
    {
      "x-us": 20000,
      "y-count": 0
    },
    {
      "x-us": 25000,
      "y-count": 0
    },
    {
      "x-us": 30000,
      "y-count": 0
    },
    {
      "x-us": 35000,
      "y-count": 0
    },
    {
      "x-us": 40000,
      "y-count": 0
    },
    {
      "x-us": 45000,
      "y-count": 0
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
qos	Qos值的组合 0x0001: sync 0x0002: not sync 0x0004: Underflow 0x0008: Overflow 0x0010: Late 0x0020: unmute 0x0040: mute
duration	sink运行时间, 单位s
peak	sink最大延时, 单位us
average	sink平均延时, 单位us
late	sink超过max-delay的数量
lost	lost packets
buff-count	sink buff 数量
buff-underflow	sink buff 下溢次数
buff-overflow	sink buff 上溢次数
statistics[i].x-us	x坐标, 时间, 单位us
statistics[i].y-count	y坐标, 数量

## /api/aes67/sink-statistics-clear 接口

清空sink 统计信息

### 请求方式

```
POST /api/aes67/sink-statistics-clear
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
sink-idx	sink组编号, 范围[0,3]

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/aes67/sdp-parse 接口

SDP解析

## 请求方式

POST /api/aes67/sdp-parse

属性	说明
sdp	SDP信息

## 返回数据

```
{  
  "status": 0  
  ...  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
source-src-ip	source 源IP信息
source-src-mac	source 源Mac信息
source-dst-ip	source 目标地址
source-dst-port	source 目标端口
ssrc	source RTP SSRC
sample-rate	设置采样率, 设备目前仅支持44100, 48000, 96000
ptime	packet time, 支持0.125ms, 0.250ms, 0.333ms, 1ms, 4ms
depth	采样率 L16, L24
channel-num	设备通道数
mediaclk-offset	RTP offset信息
ttl	TTL值, 范围[1, 255]

# /api/ices/receiver-select 接口

Stream RX流选择

## 请求方式

POST /api/ices/receiver-select

属性	说明
type	0: NDI 1:SRT 2:icecast

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。



# /api/ndi/sender-info 接口

NDI TX端信息

## 请求方式

POST /api/ndi/sender-info

## 返回数据

```
{
  "status": 0,
  "discovery-server": "",
  "ndi": [
    {
      "uid": 1,
      "enable": true,
      "ndi-name": "D424211217005_0_01",
      "group": "public",
      "enable-full": true,
      "discovery-server-enable": true,
      "audio-standard": 0,
      "mode": 3,
      "multicast-ttl": 0,
      "multicast-net-prefix": "",
      "multicast-netmask": "",
      "failover-enable": false,
      "failover-ndi-name": "",
      "failover-url": "",
      "web-control": true,
      "report": {
        "module-name": "mws_ndi_sink_0",
        "module-type": 34,
        "ndi-name": "PRO-CONVERT-AES67 (D424211217005_0_01)",
        "num-clients": 0
      }
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
discovery-server	NDI discovery server 地址
ndi[i].uid	唯一ID, 非0
ndi[i].enable	使能NDI TX
ndi[i].ndi-name	NDI名称
ndi[i].group	NDI组名称
ndi[i].enable-full	使能NDI FULL, 默认NDI HX
ndi[i].discovery-server-enable	使能discovery server
ndi[i].audio-standard	音量标准 0: SMPTE 1: EBU
ndi[i].mode	传输方式: 0:UDP 单播 1:UDP 组播 2:RUDP 单播 3:TCP 单连接 4:TCP 多连接
ndi[i].multicast-ttl	UDP 组播模式下TTL值
ndi[i].multicast-net-prefix	UDP 组播模式下IP地址
ndi[i].multicast-netmask	UDP 组播模式下子网掩码
ndi[i].failover-enable	使能failover

ndi[i].failover-ndi-name	failover功能, 被选择的NDI TX名称
ndi[i].failover-url	failover功能, 被选择的NDI TX的URL
ndi[i].web-control	true/false
ndi[i].report	运行报告

# /api/ndi/sender-apply 接口

NDI TX端设置

## 请求方式

POST /api/ndi/sender-apply

属性	说明
uid	唯一ID, 非0
enable	使能NDI TX
ndi-name	NDI名称
group	NDI组名称
enable-full	使能NDI FULL, 默认NDI HX
discovery-server-enable	使能discovery server
mode	传输方式： 0:UDP 单播 1:UDP 组播 2:RUDP 单播 3:TCP 单连接 4:TCP 多连接
audio-standard	音量标准 0: SMPTE 1: EBU
multicast-ttl	UDP 组播模式下TTL值
multicast-net-prefix	UDP 组播模式下IP地址
multicast-netmask	UDP 组播模式下子网掩码
failover-enable	使能failover
failover-ndi-name	failover功能, 被选择的NDI TX名称
failover-url	failover功能, 被选择的NDI TX的URL
web-control	true/false

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/ndi/sender-del 接口

NDI TX端删除

## 请求方式

POST /api/ndi/sender-del

属性	说明
uid	唯一ID, 非0

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /api/ndi/sender-find 接口

NDI TX端设备发现

### 请求方式

POST /api/ndi/sender-find

### 返回数据

```
{
  "status": 0,
  "sources": [
    {
      "ndi-name": "DESKTOP-KN2V7CQ (Intel UHD Graphics 630 1)",
      "url": "10.10.14.202:5961"
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
ndi[i].ndi-name	NDI 名称
ndi[i].url	NDI 设备URL

# /api/ndi/sender-options 接口

NDI TX端公共项设置

## 请求方式

POST /api/ndi/sender-options

属性	说明
discovery-server	格式server1, server2...

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/ndi/receiver-info 接口

NDI RX端信息

## 请求方式

POST /api/ndi/receiver-info

## 返回数据

```
{
  "status": 0,
  "discovery-server": "",
  "group": "",
  "extra-ips": "",
  "ndi": [
    {
      "uid": 1,
      "enable": true,
      "is-find": true,
      "alias": "DESKTOP-KN2V7CQ (Intel UHD Graphics 630 1)",
      "ndi-name": "DESKTOP-KN2V7CQ (Intel UHD Graphics 630 1)",
      "url": "10.10.14.202:5961",
      "audio-standard": 0,
      "buffer-duration": 60,
      "headroom-db": -20,
      "format": {
        "sample-rate": 48000,
        "channel-num": 2,
        "bit-depth": 32
      },
      "report": {
        "audio-jitter": 22222,
        "audio-kbps": 3000,
        "is-connected": true,
        "living-time-ms": 6442306,
        "module-name": "mws_ndi_src_0",
        "module-type": 33,
        "video-jitter": 0,
        "video-kbps": 0
      }
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
discovery-server	Discovery Server 信息
group	组信息
extra-ips	额外IP地址
ndi[i].uid	唯一ID, 非0
ndi[i].enable	使能true/false
ndi[i].is-find	NDI 是否来源设备发现
ndi[i].alias	别名
ndi[i].ndi-name	NDI 名称
ndi[i].url	NDI 设备URL
ndi[i].audio-standard	音量标准 0: SMPTE 1: EBU
ndi[i].buffer-duration	缓存时间单位ms, 有效值[1,120]
ndi[i].headroom-db	Headroom, 单位dB

ndi[i].format.sample-rate	采样率
ndi[i].format.channel-num	通道数
ndi[i].format.bit-depth	位宽
ndi[i].report	运行报告



# /api/ndi/receiver-apply 接口

NDI RX端设置

## 请求方式

POST /api/ndi/receiver-apply

属性	说明
uid	唯一ID, 非0
enable	使能true/false
is-find	NDI 是否来源设备发现
alias	别名
ndi-name	NDI 名称
url	NDI 设备URL
audio-standard	音量标准 0: SMPTE 1: EBU
buffer-duration	缓存时间单位ms, 有效值[1,120]
headroom-db	Headroom, 单位dB

## 返回数据

```
{  
  "status": 0
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/ndi/receiver-del 接口

NDI RX端删除

## 请求方式

```
POST /api/ndi/receiver-del
```

属性	说明
uid	唯一ID, 非0

## 返回数据

```
{  
  "status": 0
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /api/ndi/receiver-find 接口

NDI RX端设备发现

### 请求方式

POST /api/ndi/receiver-find

### 返回数据

```
{
  "status": 0,
  "sources": [
    {
      "ndi-name": "DESKTOP-KN2V7CQ (Intel UHD Graphics 630 1)",
      "url": "10.10.14.202:5961"
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
ndi[i].ndi-name	NDI 名称
ndi[i].url	NDI 设备URL

## /api/ndi/receiver-options 接口

NDI RX端公共项设置

### 请求方式

```
POST /api/ndi/receiver-options
```

属性	说明
discovery-server	格式server1, server2...
group	组信息
extra-ips	额外IP地址

### 返回数据

```
{  
  "status": 0
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/srt/sender-info 接口

SRT TX端信息

## 请求方式

POST /api/srt/sender-info

## 返回数据

```
{
  "status": 0,
  "srt": [
    {
      "uid": 1,
      "enable": true,
      "name": "srt_01",
      "mode": 1,
      "remote-ip": "",
      "remote-port": 8000,
      "bind-port": 10000,
      "stream-id": "1",
      "connect-timeout": 3000,
      "retry-duration": 3000,
      "latency": 120,
      "bandwidth": 25,
      "mtu": 1500,
      "enc": 0,
      "passphrase": "",
      "report": {
        "living-time-ms": 8970,
        "mode": "listener",
        "module-name": "mws_srt_sink_0",
        "module-type": 114,
        "num-clients": 0
      }
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
srt[i].uid	唯一ID, 非0
srt[i].enable	使能true/false
srt[i].name	名称
srt[i].mode	SRT 模式: 0:caller 1:listener
srt[i].remote-ip	caller 模式IP地址
srt[i].remote-port	caller 模式端口
srt[i].bind-port	listener模式绑定端口
srt[i].stream-id	Stream ID
srt[i].connect-timeout	连接超时, 单位ms
srt[i].retry-duration	重试等待时间, 单位ms
srt[i].latency	延时时间, 单位ms
srt[i].bandwidth	带宽, 百分比
srt[i].mtu	MTU值, 【0, 1500】
srt[i].enc	加密方式, 0: 不加密 16: aes-128 24: aes-192 32: aes-256
srt[i].passphrase	秘钥



# /api/srt/sender-apply 接口

SRT TX端设置

## 请求方式

POST /api/srt/sender-apply

属性	说明
uid	唯一ID, 非0
enable	使能true/false
name	名称
mode	SRT 模式: 0:caller 1:listener
remote-ip	caller 模式IP地址
remote-port	caller 模式端口
bind-port	listener模式绑定端口
stream-id	Stream ID
connect-timeout	连接超时, 单位ms
retry-duration	重试等待时间, 单位ms
latency	延时时间, 单位ms
bandwidth	带宽, 百分比
mtu	MTU值, 0 到 1500
enc	加密方式, 0: 不加密 16: aes-128 24: aes-192 32: aes-256
passphrase	密钥

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/srt/sender-del 接口

SRT TX端删除

## 请求方式

POST /api/srt/sender-del

属性	说明
uid	唯一ID, 非0

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。



# /api/srt/receiver-info 接口

SRT RX端信息

## 请求方式

POST /api/srt/receiver-info

## 返回数据

```
{
  "status": 0,
  "srt": [
    {
      "uid": 1,
      "enable": true,
      "name": "srt_01",
      "mode": 1,
      "remote-ip": "",
      "remote-port": 8000,
      "bind-port": 8000,
      "stream-id": "1",
      "latency": 120,
      "is-encrypted": false,
      "passphrase": "",
      "buffer-duration": 120,
      "headroom-db": 0,
      "format": {
        "sample-rate": 0,
        "channel-num": 0,
        "bit-depth": 0
      },
      "report": {}
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
srt[i].uid	唯一ID, 非0
srt[i].enable	使能true/false
srt[i].name	名称
srt[i].mode	SRT 模式: 0:caller 1:listener
srt[i].remote-ip	caller 模式IP地址
srt[i].remote-port	caller 模式端口
srt[i].bind-port	listener模式绑定端口
srt[i].stream-id	Stream ID
srt[i].latency	延时时间, 单位ms
srt[i].mtu	MTU值[0,1500]
srt[i].is-encrypted	启用加密true/false
srt[i].passphrase	秘钥
srt[i].buffer-duration	缓存时间单位ms, 有效值[1,120]
srt[i].headroom-db	Headroom, 单位dB
srt[i].format.sample-rate	采样率
srt[i].format.channel-num	通道数
srt[i].format.bit-depth	位宽



# /api/srt/receiver-apply 接口

SRT RX端设置

## 请求方式

POST /api/srt/receiver-apply

属性	说明
uid	唯一ID, 非0
enable	使能true/false
name	名称
mode	SRT 模式: 0:caller 1:listener
remote-ip	caller 模式IP地址
remote-port	caller 模式端口
bind-port	listener模式绑定端口
stream-id	Stream ID
latency	延时时间, 单位ms
mtu	MTU值[0,1500]
is-encrypted	启用加密true/false
passphrase	秘钥
buffer-duration	缓存时间单位ms, 有效值[1,1000]
headroom-db	Headroom, 单位dB

## 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/srt/receiver-del 接口

SRT RX端删除

## 请求方式

POST /api/srt/receiver-del

属性	说明
uid	唯一ID, 非0

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /api/ices/sender-info 接口

icecast TX端信息

### 请求方式

POST /api/ices/sender-info

### 返回数据

```
{
  "status": 0,
  "ices": [
    {
      "uid": 1,
      "enable": true,
      "name": "test1",
      "service-url": "http://110.40.192.158:8000",
      "username": "source",
      "password": "hackme",
      "state": "success"
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
uid	唯一ID, 非0
enable	使能
name	icecast 流名称
service-url	icecast 服务器URL
username	icecast 服务器推流用户名
password	icecast 服务器推流用户密码

# /api/ices/sender-apply 接口

icecast TX端设置

## 请求方式

POST /api/ices/sender-apply

属性	说明
uid	唯一ID, 非0
enable	使能
name	icecast 流名称
service-url	icecast 服务器URL
username	icecast 服务器推流用户名
password	icecast 服务器推流用户密码

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/ices/sender-del 接口

icecast TX端删除

## 请求方式

POST /api/ices/sender-del

属性	说明
uid	唯一ID, 非0

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/ices/receiver-info 接口

icecast RX端信息

## 请求方式

POST /api/ices/receiver-info

## 返回数据

```
{
  "status": 0,
  "ices": [
    {
      "uid": 1,
      "enable": true,
      "name": "icecast_01",
      "service-url": "http://ec5.yesstreaming.net:1930",
      "room": "stream",
      "buffer-duration": 500,
      "headroom-db": 0,
      "state": "fail",
      "format": {
        "sample-rate": 0,
        "channel-num": 0,
        "bit-depth": 0,
        "coding-format": "NONE"
      }
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
ices[i].uid	唯一ID, 非0
ices[i].enable	使能
ices[i].name	名称
ices[i].service-url	icecast 服务器URL
ices[i].room	icecast流名称
ices[i].buffer-duration	缓存设置, 单位ms, 范围[1,1000]
ices[i].headroom	headroom, 单位dB
ices[i].state	连接状态: idle, connecting, success, connect-fail, auth-fail, recv-fail, fail
ices[i].format.sample-rate	采样率
ices[i].format.channel-num	通道数
ices[i].format.bit-depth	位宽
ices[i].format.codeing-format	ices格式: NONE, PCM, AAC, MP3



# /api/ices/receiver-apply 接口

icecast RX端设置

## 请求方式

POST /api/ices/receiver-apply

属性	说明
uid	唯一ID, 非0
enable	使能
name	名称
service-url	icecast 服务器URL
room	icecast流名称
buffer-duration	缓存设置, 单位ms, 范围[1,1000]
headroom	headroom, 单位dB

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/ices/receiver-del 接口

icecast RX端删除

## 请求方式

POST /api/ices/receiver-del

属性	说明
uid	唯一ID, 非0

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /api/ices/list-streams 接口

获取icecast服务所有流信息

### 请求方式

POST /api/ices/list-streams

### 返回数据

```
{
  "streams": [
    {
      "room": "icecast_02",
      "content-type": "audio/aac",
      "is-magewell": true
    },
    {
      "room": "test1",
      "content-type": "audio/aac",
      "is-magewell": true
    }
  ],
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
streams[i].room	流名称
streams[i].content-type	流的格式
streams[i].is-magewell	是否属于magewell

# /api/aes67/matrix-info 接口

AES67音频矩阵信息

## 请求方式

POST /api/aes67/matrix-info

## 返回数据

```
{
  "status": 0,
  "max-tx-channels": 22,
  "max-rx-channels": 46,
  "matrix": [
    {
      "tx-no": 0,
      "tx-mute": false,
      "tx-volume": 0,
      "mix-state": [
        ],
      "rx-mute": [
        ],
      "rx-volume": [
        ]
    },
    {
      "tx-no": 1,
      "tx-mute": false,
      "tx-volume": 0,
      "mix-state": [
        ],
      "rx-mute": [
        ],
      "rx-volume": [
        ]
    },
    {
      "tx-no": 2,
      "tx-mute": false,
      "tx-volume": 0,
      "mix-state": [
        ],
      "rx-mute": [
        ],
      "rx-volume": [
        ]
    },
    {
      "tx-no": 3,
      "tx-mute": false,
      "tx-volume": 0,
      "mix-state": [
        ],
      "rx-mute": [
        ]
    },
  ],
}
```

```

    "rx-volume": [
    ]
  },
  {
    "tx-no": 4,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  },
  {
    "tx-no": 5,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  },
  {
    "tx-no": 6,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  },
  {
    "tx-no": 7,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  },
  {
    "tx-no": 8,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  }
]

```

```

    },
    {
      "tx-no": 9,
      "tx-mute": false,
      "tx-volume": 0,
      "mix-state": [

      ],
      "rx-mute": [

      ],
      "rx-volume": [

      ]
    },
    {
      "tx-no": 10,
      "tx-mute": false,
      "tx-volume": 0,
      "mix-state": [

      ],
      "rx-mute": [

      ],
      "rx-volume": [

      ]
    },
    {
      "tx-no": 11,
      "tx-mute": false,
      "tx-volume": 0,
      "mix-state": [

      ],
      "rx-mute": [

      ],
      "rx-volume": [

      ]
    },
    {
      "tx-no": 12,
      "tx-mute": false,
      "tx-volume": 0,
      "mix-state": [

      ],
      "rx-mute": [

      ],
      "rx-volume": [

      ]
    },
    {
      "tx-no": 13,
      "tx-mute": false,
      "tx-volume": 0,
      "mix-state": [

      ],
      "rx-mute": [

      ],
      "rx-volume": [

      ]
    },
    {
      "tx-no": 14,

```

```

    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  },
  {
    "tx-no": 15,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  },
  {
    "tx-no": 16,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  },
  {
    "tx-no": 17,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  },
  {
    "tx-no": 18,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

    ],
    "rx-mute": [

    ],
    "rx-volume": [

    ]
  },
  {
    "tx-no": 19,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [

```

```

    ],
    "rx-mute": [
    ],
    "rx-volume": [
    ]
  },
  {
    "tx-no": 20,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [
    ],
    "rx-mute": [
    ],
    "rx-volume": [
    ]
  },
  {
    "tx-no": 21,
    "tx-mute": false,
    "tx-volume": 0,
    "mix-state": [
    ],
    "rx-mute": [
    ],
    "rx-volume": [
    ]
  }
],
"uac-rx": {
  "channel-num": 4
},
"live-rx": {
  "type": 0,
  "lives": [
    {
      "type": 0,
      "name": "DESKTOP-KN2V7CQ (Intel UHD Graphics 630 1)",
      "channel-num": 2
    }
  ]
},
"aes67-rx": [
  {
    "sink-idx": 0,
    "name": "Pro_Convert_AES67_1015_0_01",
    "channel-num": 2
  }
],
"uac-tx": {
  "channel-num": 4
},
"live-tx": {
  "ndi": [
    "D424211217005_0_01"
  ],
  "srt": [],
  "ices": []
},
"aes67-tx": [
  {
    "name": "Pro_Convert_AES67_7005_0_01",
    "channel-num": 2
  }
]

```



```
]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
max-tx-channels	设备支持最大的TX 通道数
max-rx-channels	设备支持最大的RX 通道数
matrix[i].tx-no	TX 通道编号
matrix[i].tx-mute	TX 静音 值false/true
matrix[i].tx-volume	TX 端音量 [-36, 36], 单位dB
matrix[i].mix-state	TX-RX连接状态: 0: 未连接 1: 连接成功 2:连接中 3: 连接失败
matrix[i].rx-mute	TX-RX 静音 值false/true
matrix[i].rx-volume	TX-RX音量 [-36, 36], 单位dB
uac-rx.channel-num	UAC RX 通道数
live-rx.type	流选择 0: NDI 1:SRT 2:icecast
live-rx.lives[i].type	流类型 0: NDI 1:SRT 2:icecast
live-rx.lives[i].name	流名称
live-rx.lives[i].channel-num	流通道数
aes67-rx.sink-idx	AES67 sink 唯一ID
aes67-rx.name	AES67 sink 名称
uac-tx.channel-num	UAC TX 通道数
live-tx.ndi	ndi流发送名称列表
live-tx.srt	SRT流发送名称列表
live-tx.icecast	icecast流发送名称列表
aes67-rx.channel-num	aes67 sink 设备通道数
aes67-tx.name	aes67 source 名称
aes67-tx.channel-num	aes67 source 设备通道数





# /api/aes67/audio-meter-limit 接口

获取AES67矩阵指定音量信息

## 请求方式

POST /api/aes67/audio-meter-limit

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
tx-no	TX 通道号
rx-no	RX 通道号

e.g.

```
{
  "tx-no": 1,
  "rx-no": [1, 2, 3, 4]
}
```

## 返回数据

```
{
  "tx-dBFS": -100,
  "tx-min-db": -100,
  "tx-max-db": 40,
  "rx-channels": [
    {
      "rx-no": 1,
      "dBFS": -87,
      "min-db": -100,
      "max-db": 40
    },
    {
      "rx-no": 2,
      "dBFS": -91,
      "min-db": -100,
      "max-db": 40
    },
    {
      "rx-no": 3,
      "dBFS": -88,
      "min-db": -100,
      "max-db": 40
    },
    {
      "rx-no": 4,
      "dBFS": -100,
      "min-db": -100,
      "max-db": 40
    }
  ],
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
tx-dBFS	TX通道dBFS值
tx-min-db	TX通道dBFS最小值
tx-max-db	TX通道dBFS最大值
rx-channels[i].rx_no	RX 通道号

rx-channels[i].dBFS	RX通道dBFS值
rx-channels[i].min-db	RX通道号dBFS最小值
rx-channels[i].max-db	RX通道号dBFS最大值

## /api/aes67/matrix-meter 接口

获取AES67矩阵指定通道音量信息

### 请求方式

POST /api/aes67/matrix-meter

属性	说明
tx-no	TX 通道编号
rx-no	RX 通道编号

### 返回数据

```
{
  "status": 0,
  "dBFS": -100,
  "min-db": -100,
  "max-db": 40
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
dbfs	dBFS的值
min-db	dBFS最小值
max-db	dBFS最大值

# /api/aes67/matrix-settings 接口

设置音频矩阵

## 请求方式

POST /api/aes67/matrix-settings

属性	说明
matrix[i].tx-no	TX 通道编号
matrix[i].rx-no	RX 通道编号
matrix[i].mix	TX-RX连接状态, 0: 未连接, 1: 连接

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /api/aes67/matrix-clear 接口

清空音频矩阵设置

## 请求方式

```
POST /api/aes67/matrix-clear
```

## 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。



## /api/aes67/volume-tx 接口

设置音频矩阵TX音量

### 请求方式

POST /api/aes67/volume-tx

属性	说明
volumes[i].tx-no	TX 通道号
volumes[i].tx-mute	TX 静音 值false/true
volumes[i].db	音量 [-36, 36], 单位dB

### 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /api/aes67/volume-mix 接口

设置音频矩阵TX-RX音量

### 请求方式

POST /api/aes67/volume-tx

属性	说明
volumes[i].tx-no	TX 通道号
volumes[i].rx-no	RX 通道号
volumes[i].rx-mute	静音
volumes[i].db	音量 [-36, 36], 单位dB

### 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /system/device-info 接口

获取设备信息。请注意capability各子项是否为true，当true时相应的API访问才有效

### 请求方式

POST /api/system/device-info

### 返回数据

```
{
  "device-name": "USB Fusion",
  "product-id": "0x506",
  "product-name": "USB Fusion",
  "hardware-rev": "A",
  "serial-number": "A506210323002",
  "firmware-ver": "1.1.202",
  "firmware-name": "Development",
  "build-time": "2021-12-17 01:07:22",
  "capability": {
    "support-timezone": true,
    "support-ntp": true,
    "support-4g": false,
    "support-station": true,
    "support-ap": true,
    "support-online-upgrade": true,
    "support-sc-control": true,
    "support-ipv6": false
  },
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
product-id	设备id。
product-name	设备名称。
hardware-rev	硬件版本。
serial-number	设备序列号。
firmware-ver	设备固件版本。
firmware-name	设备固件名称。
build-time	设备固件构建时间。
capability.support-timezone	支持时区。
capability.support-ntp	支持NTP。
capability.support-4g	4G模块。
capability.support-station	WIFI 支持STA模式。
capability.support-ap	WIFI 支持AP模式。
capability.support-online-upgrade	支持在线升级。
capability.support-sc-control	支持云管理。
capability.support-ipv6	支持IPv6。

## /system/info 接口

获取cpu, 内存信息等信息。

### 请求方式

POST /api/system/info

### 返回数据

```
{
  "device-name": "USB Fusion",
  "uptime": 8410,
  "cpu": {
    "total": 1624896,
    "idle": 1281701,
    "usage": 2110
  },
  "mem": {
    "total": 8069612,
    "avail": 7171768
  },
  "datetime": {
    "cur-time": "2021-12-20 13:25:57",
    "zonename": "Asia/Shanghai",
    "ntp-enable": true,
    "ntp-server1": "0.pool.ntp.org",
    "ntp-server2": "1.pool.ntp.org"
  },
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
device-name	设备名称。
uptime	开机时长, 单位s。
cpu.total	CPU 总时间。
cpu.idle	CPU 空闲时间。
cpu.usage	CPU 使用率 x 100。
mem.total	系统总内存, 单位KB。
mem.avail	系统可用内存, 单位KB。
datetime.cur-time	系统时间, 格式: yyyy-MM-dd HH:mm:ss。
datetime.zonename	时区名称。
datetime.ntp-enable	NTP 使能。
datetime.ntp-server1	NTP服务器1。
datetime.ntp-server2	NTP服务器2。

## /system/set-device-name 接口

设置设备名称。

### 请求方式

POST /api/system/set-device-name

参数	说明
name	设备名称。

### 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /system/set-date-time 接口

NTP 功能设置。

### 请求方式

POST /api/system/set-date-time

参数	说明
ntp-enable	是否使能NTP。
ntp-server1	NTP 服务器1。
ntp-server2	NTP 服务器2。
time	本地时间，格式：yyyy-MM-dd HH:mm:ss。

### 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0：执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /system/timezone-set 接口

时区设置。

### 请求方式

```
POST /api/system/timezone-set
```

参数	说明
zonename	时区名称。

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /network/if-info 接口

网卡信息。

### 请求方式

POST /api/network/if-info

### 返回数据

```
{
  "device-name": "USB Fusion yxy1",
  "net": [
    {
      "enable": true,
      "iface": "eth0",
      "type": 0,
      "use-dhcp": true,
      "ipaddr": "10.10.12.166",
      "netmask": "255.255.240.0",
      "gateway": "10.10.0.1",
      "mac": "84:85:86:87:88:2e",
      "link-speed": 1000,
      "link-state": 2,
      "tx-speed-kbps": 0,
      "rx-speed-kbps": 107
    },
    {
      "enable": true,
      "iface": "wlan0",
      "type": 1,
      "mode": 1,
      "ssid": "USB-Fusion_yx_5G",
      "use-dhcp": true,
      "ipaddr": "192.168.67.1",
      "netmask": "255.255.255.0",
      "gateway": "",
      "mac": "10:2c:6b:fd:9b:78",
      "link-speed": -1,
      "link-state": 2,
      "tx-speed-kbps": 3,
      "rx-speed-kbps": 0
    },
    {
      "enable": true,
      "iface": "usb0",
      "type": 3,
      "use-dhcp": true,
      "ipaddr": "192.168.66.1",
      "netmask": "255.255.255.0",
      "gateway": "192.168.66.1",
      "mac": "8e:40:df:be:7c:fa",
      "link-speed": 480,
      "link-state": 2,
      "tx-speed-kbps": 0,
      "rx-speed-kbps": 0
    }
  ],
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
device-name	设备名称。
net[i].enable	网卡服务是否开启。



net[i].iface	网卡名称。
net[i].type	网卡类型 0: 以太网 1: WIFI 2: 4G 模块 3: USB。
net[i].mode	WIFI 工作模式, 当net[i].type == 1 存在, 0: STA 模式 1: AP 模式。
net[i].ssid	WIFI ssid。
net[i].reboot-require	WIFI 重启生效。
net[i].use-dhcp	true : 使用dhcp获取ip false: static 配置网络。
net[i].ipaddr	IP 地址。
net[i].netmask	子网掩码。
net[i].ipv6addr	IPv6 地址信息。
net[i].gateway	网关地址。
net[i].mac	MAC 地址。
net[i].link-speed	速率 10 : 10Mbps, 100: 100Mbps, 1000: 1Gbps, 2500: 2.5Gbps, 10000: 10Gbps。 usb 支持的速率 12: full-speed, 480: high-speed, 5000: super-speed-5g, 10000: super-speed-10g
net[i].link-state	端口状态 0: down, 1: disconnected, 2: connected
net[i].vendor	4G模块厂商信息。
net[i].product	4G模块型号信息。
net[i].tx-speed-kbps	发送速度 (Kbps)。
net[i].rx-speed-kbps	接收速度 (Kbps)。

## /network/if-set 接口

配置网卡。

### 请求方式

POST /api/network/if-set

参数	说明
iface	网卡名称。
use-dhcp	true: 使用DHCP获取IP; false: static 配置网络
ipaddr	IP 地址, 当 use-dhcp 为 false 需带上
netmask	子网掩码, 当 use-dhcp 为 false 需带上
gateway	网关地址, 当 use-dhcp 为 false 需带上

### 返回数据

```
{
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /network/if-route 接口

获取默认路由。

### 请求方式

```
POST /api/network/if-route
```

### 返回数据

```
{
  "iface": "",
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
ifname	默认路由经过的网卡, iface 若为空, 表示暂时无路由。

## /network/get-dns 接口

获取DNS。

### 请求方式

```
POST /api/network/get-dns
```

### 返回数据

```
{
  "is-manual": false,
  "dns1": "10.0.1.3",
  "dns2": "",
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
is-manual	是否手动设置DNS。
dns1	DNS, 空字符表示未设置。
dns2	DNS, 空字符表示未设置。

## /network/set-dns 接口

设置DNS。

### 请求方式

POST /api/network/set-dns

参数	说明
is-manual	是否手动设置DNS。
dns1	DNS，空字符表示未设置。
dns2	DNS，空字符表示未设置。

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /network/usb-config 接口

配置USB网卡。

### 请求方式

```
POST /api/network/usb-config
```

参数	说明
iface	网卡名称。
ipaddr	IP 地址。

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /user/login 接口

用户登录，登录成功后会在 Cookie 存放 Session ID (Cookie: sid-[serial number]=t2i704wbvoy51y408p588bpji010ibp0)。

### 请求方式

POST /api/user/login

参数	说明
username	用户名。
password	密码，使用 sha256加密。

### 返回数据

```
{
  "status": 0,
  "sid": "t2i704wbvoy51y408p588bpji010ibp0"
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

### 接口示例

```
// login (username: Admin, password=Admin)
curl --cookie-jar sid.txt http://192.168.66.1/api/user/login -X POST -H 'Content-Type: application/json' -d '{"username": "Admin", "password": "c1c224b03cd9bc7b6a86d77f5dace40191766c485cd55dc48caf9ac873335d6f"}'
```

## /user/logout 接口

退出登录，返回到登录界面。

### 请求方式

```
POST /api/user/logout
```

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。



## /user/get-all 接口

获取系统用户列表信息，仅管理员有权限。

### 请求方式

```
POST /api/user/get-all
```

### 返回数据

```
{
  "users": [
    {
      "username": "Admin",
      "group": "Admin"
    }
  ],
  "status": 0
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
users	用户列表数组。username: 用户名, group: 用户组。

## /user/add 接口

添加用户，仅管理员有权限。

### 请求方式

POST /api/user/add

参数	说明
username	用户名。
password	密码，使用 sha256加密。

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /user/del 接口

删除用户，仅管理员有权限。

### 请求方式

POST /api/user/del

参数	说明
username	用户登录名。

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /user/ch-password 接口

用户修改自己的登录密码，修改时必须输入原密码。

### 请求方式

POST /api/user/ch-password

参数	说明
password	原密码，使用 sha256加密。
new-password	新密码，使用 sha256加密。

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /user/set-password 接口

重置用户密码，无需输入原密码，仅管理员有权限。

### 请求方式

POST /api/user/set-password

参数	说明
username	用户登录名。
password	新密码，使用 sha256加密。

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## upgrade/online-check 接口

启动在线升级检查。

### 请求方式

```
POST /api/upgrade/online-check
```

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# upgrade/online-check-result 接口

获取在线检查结果。

## 请求方式

```
POST /api/upgrade/online-check-result
```

## 返回数据

```
{  
  "up-to-date": true,  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
up-to-date	当前版本是否为最新，有效值：true/false。
version	最新版本。
size	最新版本的大小。
md5	最新版本的md5值。
changeLog	最新版本升级内容

## /upgrade/upload-fw 接口

上传固件，上传文件格式必须为.mwf，必须使用POST multipart/form-data上传文件。

### 请求方式

POST /upgrade/upload-fw

### 返回数据

```
{
  "status": 0,
  "up-to-date": true,
  "version": "1.1.72"
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
up-to-date	是否是最新版。
version	上传的固件版本号。



## /upgrade/update 接口

执行更新操作，更新过程中可以通过 [/upgrade/state](#) 接口获取当前状态。

### 请求方式

POST /api/upgrade/update

参数	说明
is-online	false: 离线升级, true: 在线升级
mode	升级模式选择 0: Auto 模式, 自动选择 Upgrade/Factory/FactoryClear 模式 1: Upgrade 模式 2: Factory 模式 3: FactoryClear 模式
timeout	升级进度不变超时时间, 单位 s。

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /upgrade/state 接口

获取当前固件版本信息和升级状态，仅管理员有权限。

### 请求方式

POST /api/upgrade/state

### 返回数据

```
{
  "status": 0,
  "state": "updating",
  "cur-ver": "1.1.72",
  "update-version": "1.1.72",
  "num-steps": 4,
  "step": 2,
  "step-name": "Erasing image",
  "step-progress": 28
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
state	任务执行状态，0: 空闲，1: 初始化升级，2: 升级中，3: 升级完成，4: online 固件下载中
cur-ver	当前固件版本号
update-version	最新固件版本号
step	执行到第几步，仅在 state 为 2 状态下存在。
num-steps	总计需要几个步骤，仅在 state 为 2 状态下存在。
step-name	当前执行步骤的名称，仅在 state 为 2 状态下存在。
step-progress	当前执行的步骤的进度，值为 0 ~ 100，单位%，仅在 state 为 2 状态下存在。
download-percent	在线下载百分比

## /upgrade/clear 接口

清除升级状态。

### 请求方式

```
POST /upgrade/clear
```

### 返回数据

```
{  
  "status": 0  
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

## /log/clear 接口

清除全部系统日志，仅管理员有权限。

### 请求方式

```
POST /api/log/clear
```

### 返回数据

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。

# /log/filter 接口

过滤日志。

## 请求方式

POST /api/log/filter

参数	说明
types	日志类型, 有效值: all、info、warn、error, 当有多个类型时用英文逗号隔开。
key	过滤关键字, 可以为空字符串。

## 返回数据

```
{
  "status": 0,
  "logs": [
    {
      "no": 0,
      "time": "2022/09/09 16:11:07.920",
      "type": "info",
      "message": "xxxxxxx"
    },
    {
      "no": 1,
      "time": "2022/09/09 16:11:04.721",
      "type": "info",
      "message": "xxxxxxx"
    }
  ]
}
```

属性	说明
status	0: 执行成功。返回其它值请参考 <a href="#">API 状态码</a> 。
logs[i].no	编号。
logs[i].time	日期时间。
logs[i].type	输出级别, 包括 info, warn, error。
logs[i].message	日志内容。

## /log/export 接口

导出设备当前的系统日志，导出文件为 html 格式，仅管理员有权限。

### 请求方式

POST /api/log/export

参数	说明
filename	导出的文件名称

### 请求结果

直接下载 html 格式日志文件到本地。