

Ultra Encode AIO

前言

文档简介	1.1
接口约定	1.2
API 状态码	1.3
云平台 API 状态码	1.4
设备状态掩码	1.5
设备发现协议	1.6
DEMO: Node.js	1.7
DEMO: C 语言	1.8

通用接口

get-info	2.1
get-status	2.2
get-settings	2.3
ping	2.4
get-report	2.5
export-report	2.6
get-logs	2.7
export-logs	2.8
clear-logs	2.9

通用设置

set-name	3.1
set-first-over	3.2
set-softap	3.3
set-ssdp	3.4
set-volume	3.5
set-sync-offset	3.6
add-nosignal-file	3.7
del-nosignal-file	3.8
set-nosignal-file	3.9
use-nosignal-file	3.10
set-time-zone	3.11
set-date-time	3.12
set-udp-mtu	3.13
set-analog-mic-bias	3.14

输入

set-video-mixer-config	4.1
select-input-source	4.2
get-signal-info	4.3
set-video-color	4.4
set-video-input-format	4.5

set-video-output-format	4.6
-------------------------	-----

编码

set-video	5.1
set-audio	5.2
enable-deinterlace	5.3

直播

add-server	6.1
enable-server	6.2
set-server	6.3
del-server	6.4
start-test-server	6.5
stop-test-server	6.6
clear-test-server	6.7
get-ndi-sources	6.8
set-ndi-find-config	6.9
enable-ndi-hx3	6.10

录制

add-rec-channel	7.1
enable-rec-channel	7.2
set-rec-channel	7.3
del-rec-channel	7.4
get-rec-channels	7.5
clear-rec-status	7.6
add-nas	7.7
enable-nas	7.8
set-nas	7.9
del-nas	7.10

存储设备管理

start-format-usb	8.1
start-format-sd	8.2
clear-format-usb	8.3
clear-format-sd	8.4
start-test-usb	8.5
stop-test-usb	8.6
clear-test-usb	8.7
get-media-files	8.8
del-media-files	8.9

叠加

add-image	9.1
set-image	9.2
del-image	9.3

get-images	9.4
add-surface	9.5
set-enable-surface	9.6
enable-surface	9.7
set-surface	9.8
del-surface	9.9
get-surfaces	9.10

排程

add-scheduler	10.1
enable-scheduler	10.2
set-scheduler	10.3
del-scheduler	10.4
get-schedulers	10.5

设备重置

reboot	11.1
reset-all-settings	11.2

用户登录

login	12.1
logout	12.2

用户管理

get-users	13.1
add-user	13.2
del-user	13.3
ch-passwd	13.4
set-passwd	13.5

网络设置

set-net	14.1
scan-wifi-results	14.2
connect-wifi-first	14.3
connect-wifi	14.4
cancel-connect-wifi	14.5
disconnect-wifi	14.6
forget-wifi	14.7
set-connect-wifi-auto	14.8
clear-connect-wifi	14.9
open-softap	14.10
close-softap	14.11
set-rndis-config	14.12

固件更新

update	15.1
upload-update-file	15.2
cancel-download	15.3
online-update-check	15.4
clear-upgrade	15.5
clear-check-update	15.6
set-enable-check-update	15.7

EDID

import-edid	16.1
export-edid	16.2
set-default-edid	16.3
get-edid-config	16.4
set-edid-config	16.5
get-loop-through-edid	16.6
export-loop-through-edid	16.7

云平台

cloud-reg-ex	17.1
cloud-unreg-ex	17.2
cloud-status	17.3

文档简介

针对 Ultra Encode 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：

- Ultra Encode AIO

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 GET 方式, 文件上传用 POST 方式
- 请求 URL 格式: http://IP/usapi?method=xxx¶m1=value1¶m2=value2...
- 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

二、返回 JSON 数据格式

格式如下, JSON 对象中的 result 属性为 [API 状态码](#), 为 0 时表示数据获取或操作成功, 否则为其它状态码。

```
{
  "result": 0,
  "cur-status": 65552,
  "last-rec-status": 0,
  "cur-time": 0,
  "box-name": "Ultra Encode B313221201001",
  ...
}
```

API 状态码

```
{
  retSendWaiting          = 31,          // 保留
  retLivingAuthErr        = 30,          // 直播连接状态: 认证错误
  retLivingNotset         = 29,          // 未设置直播地址
  retLivingDNS            = 28,          // 直播连接状态: DNS 解析
  retInit                  = 27,          // 初始化状态
  retLivingAuthing        = 25,          // 直播连接状态: 正在授权中
  retLivingWaiting        = 24,          // 直播连接状态: 等待连接
  retLivingConnecting     = 23,          // 直播连接状态: 正在连接服务器
  retLivingConnected      = 22,          // 直播连接状态: 服务器已连接上
  retPushReboot           = 21,
  retAudioSignalChange    = 20,
  retBlueWrite            = 19,
  retBlueRead             = 18,
  retBlueShutDown        = 17,
  retDiskOn               = 16,
  retDiskOff              = 15,
  retDiskChange           = 14,
  retSnapshotOver        = 13,
  retPushReset            = 12,
  retPushLiving           = 11,
  retPushRecord           = 10,
  retSignalChange         = 9,
  retRouteChange          = 8,
  retIPChange             = 7,
  retNetChange            = 6,
  retCancel               = 5,          // 操作已取消
  retLowSpace             = 4,          // 保留
  retLowSpeed             = 3,          // 保留
  retRunning              = 2,          // 操作正在进行中
  retRepeat               = 1,          // 重复操作
  retSucceed              = 0,          // 请求成功
  errPasswd               = -1,          // 密码错误
  errOccupied             = -2,          // 设备已被占用
  errDisconnect           = -3,          // 保留
  errDevice               = -4,
  errDisk                 = -5,
  errUnconnect           = -6,
  errKey                  = -7,
  errVersion              = -8,
  errBusy                 = -9,          // 系统繁忙
  errParam                = -10,        // 请求参数有误
  errUsage                = -11,        // 保留
  errTimeout              = -12,
  errIP                   = -13,        // 保留
  errNotFound             = -14,        // 数据不存在
  errFile                 = -15,        // 文件错误
  errNoSpace              = -16,        // 保留
  errNeedAuth             = -17,        // 未登录授权
  errSystem               = -18,        // 系统错误
  errDiskSpeed            = -19,
  errEmpty                = -20,
  errNetwork              = -21,
  errEvent                = -22,
  errCodec                = -23,
  errBlue                 = -24,
  errNoUser               = -25,        // 用户不存在
  errNoPermissin         = -26,
  errSameName            = -27,        // 重名
  errString               = -28,        // 字符不合法
  errChannelsLimited      = -29,        // 最大只允许向 6 个直播服务器推流
  err8MLimited            = -30,        // 保留
  errFacebookLimited      = -31,        // 保留
  errCodecLimited         = -32,        // 保留
  err4GLimited            = -33,        // 保留
  errMWFUnsupported       = -34,        // 固件升级包与当前产品型号或硬件版本不匹配
  errNoSignal            = -35,        // 无信号
  errSDCard               = -36,
```

```

errXinYueServer      = -37,      // 保留
errAliYunOSS         = -38,      // 保留
errSDNoSpace         = -39,      // 保留
errSDNoPermission    = -40,      // 保留
errRTSPLimited       = -41,      // 当选择 RTSP 直播时, 只允许向 1 个直播服务器推流
errRTSP8MLimited     = -42,      // 保留
errBandwidthLimited  = -43,      // 保留
errPortLimited       = -44,      // 直播服务器端口已占用
errNDILimited        = -45,      // 当选择 NDI|HX 直播时, 只允许向 1 个直播服务器推流
errSRTLimited        = -46,      // 当选择 SRT Listener 直播时, 只允许向 1 个 SRT Listener 服务器推流
errNDISettings       = -47,      // 当选择 NDI|HX 直播时, 次码流最大支持 640x480@60
errSubStreamSettings = -48,      // 未选择 NDI|HX 直播时, 次码流最大支持 1280x720@30
errHLSLimited        = -49,      // 当选择 HLS 直播时, 只允许向 1 个直播服务器推流
errProtocolLimited   = -50,      // 只允许 1 种直播协议
errInit              = -51,      // 直播通道初始化失败
errDeinterlaceSettings = -52,    // 去隔行设置错误
errTVULimited        = -53,      // 当选择 TVU ISSP 直播时, 只允许向 1 个直播服务器推流
errProtocolOneChannel = -54,     // 将 errRTSPLimited/errNDILimited/errSRTLimited/errHLSLimited/errTVULi
mitted 等错误码统一
errUHDSettings       = -55,      // 主码流编码分辨率大于 2048x1080 时, 主次码流帧率不能大于 30 FPS

errInputSignal       = -56,      // 当输入信号源分辨率大于 2048x1080 时, 主次码流帧率不能大于 30 FPS
errScheduler         = -57,      // 保留
errMountPoint        = -58,      // 网络存储服务器挂载点错误
}

```


云平台 API 状态码

```
{
  errLogin      = -200,    // 未登录
  errSn         = -109,   // 非法序列号
  errParam      = -10,    // 参数错误
  errDevice     = -4,     // 云平台拒绝该类设备注册
  errPasswd     = -1,     // 邀请码错误
  retSuccess    = 0,
  retRepeat     = 1,      // 重复注册
  retRegistering = 2,     // 正在注册
  retInit       = 27,     // 未设置
  retOnline     = 35,     // 云平台在线
  retOffline    = 36,     // 云平台离线
  retDeleted    = 104,
  retWaiting    = 103,
  retRefused    = 102,
  retAccepted   = 101,
}
```

设备状态掩码

```
{
  statusFirst      = 0x01,      // 设备第一次运行
  statusRecord     = 0x02,      // 保留
  statusLiving     = 0x04,      // 保留
  statusStream     = 0x08,      // 保留
  statusDiskReady  = 0x10,      // 保留
  statusRTMPReady  = 0x20,      // 保留
  statusSoftAP     = 0x40,      // 无线网卡正处于 AP 模式下
  statusMIC        = 0x100,     // 保留
  statusPHONE      = 0x200,     // 保留
  statusOutput     = 0x400,     // 保留
  statusDiskTest   = 0x1000,    // 正在进行磁盘性能测试
  statusBlue       = 0x2000,    // 保留
  statusUpgrade    = 0x4000,    // 正在进行固件升级
  statusNetTest    = 0x8000,    // 正在进行直播服务器测试
  statusPasswd     = 0x10000,   // 设备已设置密码
  statusOccupied   = 0x20000,   // 设备锁定（最大支持两个 APP 同时连接设备）
  statusFormatDisk = 0x100000,  // 正在进行 U 盘格式化
  statusFormatSD   = 0x200000,  // 正在进行 SD 卡格式化
  statusSearchWifi = 0x400000,  // 正在搜索 Wi-Fi
  statusConnectWifi = 0x800000, // 正在连接 Wi-Fi
  statusConnectBlue = 0x1000000, // 保留
  statusCheckUpgrade = 0x2000000, // 正在检测固件信息
  statusReset      = 0x4000000, // 设备正在重置
  stausIPv6        = 0x8000000, // 保留
  statusTestLock   = 0x10000000, // 保留
  statusReboot     = 0x20000000, // 设备正在重启
}
```

设备发现协议

可以通过 **组播方式** 或 **SSDP 协议** 发现设备。

组播方式

可以获取设备的一些基本配置和状态信息，如设备名称、序列号、工作状态、网络状态等。

- 组播地址：239.255.255.250
- 端口：2538

返回数据如下：

```
{
  "version": "1.0",
  "flag": "ssip",
  "product": "Ultra Encode",
  "boxname": "Ultra Encode B313221201001",
  "serialnumber": "B313221201001",
  "wifiip": "192.168.48.1",
  "ethip": "192.168.1.217",
  "status": 65600
}
```

属性	说明
version	组播通信协议版本号
flag	协议标识
product	产品类型
boxname	设备名称
serialnumber	设备序列号
wifiip	无线网 IP 地址
ethip	以太网 IP 地址
status	设备状态掩码

DEMO: Node.js

本文通过 Node.js 环境下的两个实例介绍 Ultra Encode API 如何调用。

DEMO 下载链接: [ultra-encode-api-demo-nodejs.zip](#)

DEMO 目录结构:

```
ultra-encode-api-demo-nodejs
|
|-- httpUtils.js    // 基于 nodejs 的 HTTP 模块封装了 get 方法和 upload 方法
|-- xxxx.mwf       // upload.js 调用 upload-update-file 接口上传的测试固件
|-- get.js         // 通过 GET 方式调用接口获取数据
|-- upload.js      // 通过 POST 方式实现文件上传
```

环境准备

- 操作系统: 支持 macOS, Linux, Windows
- 运行环境: 建议选择 LTS 版本, 最低要求 8.x

运行方式

1. 在终端控制台进入 DEMO 目录

```
cd ultra-encode-demo-nodejs
```

2. 将 get.js 和 upload.js 中的 deviceIP 替换成测试设备的 IP 地址
3. 运行 get.js

```
node get
```

4. 运行 upload.js

```
node upload
```

DEMO: C 语言

环境准备

- 操作系统: 支持 Windows、macOS、Linux

源码编译

- 开发者自己准备相应平台(Windows/macOS/Linux/...) 的"curl sdk"
- DEMO 下载链接: [ultra-encode-api-demo-c.zip](#)
- 编译 "ultra_encode_curl.c", 链接到"libcurl"
- 生成可执行性文件 "ultra_encode_curl"

运行方式

- 在终端控制台进入 bin 目录, 将固件文件拷贝到相应平台, 执行相应平台的 ultra_encode_curl

```
cd ultra-encode-api/demo/c/bin
cp ultra_encode_aio_rev_b_2_0_318.mwf linux
cd linux
./ultra_encode_curl <hostip:port>
```

- 输出结果

```
***** 1. login *****
login response data:
{
  "result": 0
}

***** 2. get info *****
get info response data:
{
  "result": 0,
  "mac-addr": {
    "eth": "d0:c8:57:81:02:2d",
    "wifi": "d0:c8:57:81:02:2e",
    "softap": "d0:c8:57:81:02:2f"
  },
  "snapshot": "/tmp/sbox-snapshot/sbox-quarter.jpg",
  "main-snapshot": "/tmp/sbox-snapshot/mainstream.jpg",
  "sub-snapshot": "/tmp/sbox-snapshot/substream.jpg",
  "product": {
    "sn": "B313220727002",
    "product-id": 787,
    "hardware-ver": "B",
    "firmware-id": 1,
    "firmware-ver-s": "2.0.312",
    "factory-firmware-ver-s": "2.0.283",
    "product-name": "Ultra Encode",
    "module-name": "Ultra Encode AIO",
    "manu-name": "MAGEWELL",
    "features": 1,
    "max-lock-count": 2,
    "live-support": 4095,
    "cloud-support": 1,
    "record-support": 1,
    "scheduler-support": 1,
    "nas-support": 1,
    "multi-audio-support": 1
  },
  "nosignal": {},
  "input-source": {},
  "audio-range": {},
  "codec-cap": {},
```

```
"rec": {},
"nas": {},
"color-range": {},
"video-format": {},
"living": {},
"surface": {}
}

**** 3. upload firmware ****
upload firmware response data:
{
  "result": 0,
  "up-to-date": true,
  "version": "2.0.123",
  "size": 12494463
}
```

get-info 接口

获取设备基本信息，主要有网卡物理地址信息、产品相关基本信息、视频和音频相关参数取值范围等。

请求方式

```
GET http://ip/usapi?method=get-info
```

参数	说明
method	方法名称: get-info

返回数据

JSON 结构如下:

```
{
  "result": 0, // 返回状态
  "mac-addr": {}, // 网卡物理地址
  "snapshot": "", // 输入视频缩略图路径
  "main-snapshot": "", // 主码流视频缩略图路径
  "sub-snapshot": "", // 次码流视频缩略图路径
  "product": {}, // 产品相关信息
  "nosignal": {}, // 无信号图片相关限制
  "input-source": {}, // 输入源选择及合成相关参数取值
  "audio-range": { // 音频相关参数取值范围
    "hdmi": {},
    "linein": {},
    "lineout": {}
  },
  "codec-cap": { // 编码相关参数取值范围
    "main-stream": {}, // 主码流相关参数取值范围
    "sub-stream": {}, // 次码流相关参数取值范围
    "ndi-sub-stream": {}, // 保留
    "ndi-hx3-main-stream": {}, // NDI|HX3 主码流相关参数取值范围
    "ndi-hx3-duration": {}, // NDI|HX3 主码流帧间隔相关参数取值范围
    "resolutions": [],
    "durations": [],
    "profile": [],
    "hevc-profile": [],
    "video-kbps": [],
    "audio-streams": [], // 音频码流取值范围
    "audio-channels": [], // 音频编码通道数取值范围
    "audio-chmap": [], // 音频编码通道映射取值范围
    "audio-kbps": [],
    "gop-sec": [],
    "video-range": [],
    "stat-sec": [],
    "video-codec": [],
    "video-ar-convert-mode": []
  },
  "rec": {
    "mode": [], // 录制模式相关参数取值
    "file-prefix": [], // 录制文件名前缀相关参数取值
    "file-suffix": [], // 录制文件格式相关参数取值
    "time-unit": [],
  },
  "nas": {
    "proto": [] // 网络存储协议相关参数取值
  },
  "color-range": { // 视频色彩调节相关参数取值
    "contrast": {},
    "brightness": {},
    "saturation": {},
    "hue": {}
  }
}
"video-format": { // 视频输入输出相关参数取值
```

```

    "input-color-fmt": [],
    "output-color-fmt": [],
    "quant-range": [],
    "sat-range": []
  }
  "living": { // 直播服务相关参数取值
    "max-bandwidth": 16384,
    "ttl": {},
    "conn-timeout": {},
    "retry-duration": {},
    "srt": {
      "latency": [],
      "bandwidth": [],
      "aes": [],
      "mtu": {}
    },
    "ndi": {},
    "udp": {},
  },
  "surface": {
    "image": {},
    "type": [],
    "image-type": [],
    "condition": [],
    "position": [],
    "font-family": [],
    "font-style": [],
    "font-weight": [],
    "time-format": []
  }
}

```

返回示例

```

{
  "result": 0,
  "mac-addr": {
    "eth": "d0:c8:57:81:02:2d",
    "wifi": "d0:c8:57:81:02:2e",
    "softap": "d0:c8:57:81:02:2f"
  },
  "snapshot": "/tmp/sbox-snapshot/sbox-quarter.jpg",
  "main-snapshot": "/tmp/sbox-snapshot/mainstream.jpg",
  "sub-snapshot": "/tmp/sbox-snapshot/substream.jpg",
  "product": {
    "sn": "B313220727002",
    "product-id": 787,
    "hardware-ver": "B",
    "firmware-id": 1,
    "firmware-ver-s": "2.0.312",
    "factory-firmware-ver-s": "2.0.283",
    "product-name": "Ultra Encode",
    "module-name": "Ultra Encode AIO",
    "manu-name": "MAGEWELL",
    "features": 1,
    "max-lock-count": 2,
    "live-support": 4095,
    "cloud-support": 1,
    "record-support": 1,
    "scheduler-support": 1,
    "nas-support": 1,
    "multi-audio-support": 1
  },
  "nosignal": {
    "max-count": 4,
    "max-width": 1920,
    "max-height": 1080,
    "max-size-kb": 1024
  },
  "input-source": {
    "sources": [
      {

```



```

    "name": "Auto Scan",
    "value": 3
  },
  {
    "name": "SDI Input",
    "value": 1
  },
  {
    "name": "HDMI Input",
    "value": 2
  },
  {
    "name": "Mixer",
    "value": 4
  }
],
"video-mixer": {
  "types": [
    {
      "name": "Picture in Picture",
      "value": 0
    },
    {
      "name": "Side by Side",
      "value": 1
    }
  ],
  "pip": [
    {
      "name": "Left Top Corner",
      "value": 1
    },
    {
      "name": "Top Right Corner",
      "value": 2
    },
    {
      "name": "Left Bottom Corner",
      "value": 3
    },
    {
      "name": "Bottom Right Corner",
      "value": 4
    }
  ],
  "sbs": [
    {
      "name": "Half",
      "value": 1
    },
    {
      "name": "One Third",
      "value": 2
    },
    {
      "name": "Two Thirds",
      "value": 3
    },
    {
      "name": "One Fourth",
      "value": 4
    },
    {
      "name": "Three Fourths",
      "value": 5
    }
  ]
}
},
"audio-range": {
  "spi": {
    "max": 600,
    "min": -10000,

```

```

"def": 0
},
"linein": {
  "max": 5525,
  "min": -13900,
  "def": 0
},
"lineout": {
  "max": 600,
  "min": -5700,
  "def": 0
}
},
"codec-cap": {
  "main-stream": {
    "max-video-kbps": 32768,
    "min-video-kbps-ratio": 5,
    "max-video-kbps-ratio": 100,
    "def-video-kbps-ratio": 20,
    "min-duration": 166667,
    "max-duration": 2000000,
    "min-gop-size": 15,
    "max-gop-size": 300
  },
  "sub-stream": {
    "max-width": 1920,
    "max-height": 1200,
    "max-duration": 2000000,
    "min-duration": 166667,
    "max-video-kbps": 32768,
    "reserved": [
      {
        "w": 1920,
        "h": 1200
      }
    ]
  },
  "ndi-sub-stream": {
    "max-width": 1920,
    "max-height": 1200,
    "max-duration": 2000000,
    "min-duration": 166667,
    "max-video-kbps": 32768
  },
  "ndi-hx3-main-stream": {
    "max-width": 1920,
    "max-height": 1200,
    "max-duration": 2000000,
    "min-duration": 166667,
    "max-video-kbps": 63488,
    "min-video-kbps-ratio": 10,
    "max-video-kbps-ratio": 156,
    "def-video-kbps-ratio": 155,
    "hevc-min-video-kbps-ratio": 10,
    "hevc-max-video-kbps-ratio": 126,
    "hevc-def-video-kbps-ratio": 125,
    "min-gop-size": 2,
    "max-gop-size": 120,
    "reserved": [
      {
        "w": 1920,
        "h": 1200
      }
    ]
  },
  "ndi-hx3-durations": [
    {
      "name": "25 FPS",
      "value": 400000
    },
    {
      "name": "30 FPS",
      "value": 333333
    }
  ]
}

```

```
},
{
  "name": "50 FPS",
  "value": 200000
},
{
  "name": "60 FPS",
  "value": 166667
},
{
  "name": "Follow Input",
  "value": 0
}
],
"resolutions": [
  {
    "w": 480,
    "h": 270
  },
  {
    "w": 480,
    "h": 360
  },
  {
    "w": 640,
    "h": 360
  },
  {
    "w": 640,
    "h": 480
  },
  {
    "w": 720,
    "h": 480
  },
  {
    "w": 720,
    "h": 540
  },
  {
    "w": 720,
    "h": 576
  },
  {
    "w": 768,
    "h": 576
  },
  {
    "w": 800,
    "h": 600
  },
  {
    "w": 960,
    "h": 540
  },
  {
    "w": 1024,
    "h": 768
  },
  {
    "w": 1280,
    "h": 720
  },
  {
    "w": 1280,
    "h": 800
  },
  {
    "w": 1280,
    "h": 960
  },
  {
    "w": 1280,
```

```

    "h": 1024
  },
  {
    "w": 1440,
    "h": 900
  },
  {
    "w": 1440,
    "h": 1080
  },
  {
    "w": 1600,
    "h": 900
  },
  {
    "w": 1600,
    "h": 1200
  },
  {
    "w": 1664,
    "h": 936
  },
  {
    "w": 1920,
    "h": 1080
  },
  {
    "w": 1920,
    "h": 1200
  },
  {
    "w": 2048,
    "h": 1080
  },
  {
    "w": 3840,
    "h": 2160
  },
  {
    "w": 4096,
    "h": 2160
  }
],
"durations": [
  {
    "name": "5 FPS",
    "value": 2000000
  },
  {
    "name": "10 FPS",
    "value": 1000000
  },
  {
    "name": "15 FPS",
    "value": 666667
  },
  {
    "name": "24 FPS",
    "value": 416667
  },
  {
    "name": "25 FPS",
    "value": 400000
  },
  {
    "name": "29.97 FPS",
    "value": 333667
  },
  {
    "name": "30 FPS",
    "value": 333333
  },
  {

```

```

    "name": "50 FPS",
    "value": 200000
  },
  {
    "name": "59.94 FPS",
    "value": 166833
  },
  {
    "name": "60 FPS",
    "value": 166667
  },
  {
    "name": "Follow Input",
    "value": 0
  }
],
"profile": [
  {
    "name": "Baseline",
    "value": 0
  },
  {
    "name": "Main profile",
    "value": 1
  },
  {
    "name": "High profile",
    "value": 2
  }
],
"hevc-profile": [
  {
    "name": "Main profile",
    "value": 0
  }
],
"video-kbps": [
  {
    "name": "256 Kbps",
    "value": 256
  },
  {
    "name": "512 Kbps",
    "value": 512
  },
  {
    "name": "768 Kbps",
    "value": 768
  },
  {
    "name": "1 Mbps",
    "value": 1024
  },
  {
    "name": "1.5 Mbps",
    "value": 1536
  },
  {
    "name": "2 Mbps",
    "value": 2048
  },
  {
    "name": "3 Mbps",
    "value": 3072
  },
  {
    "name": "4 Mbps",
    "value": 4096
  },
  {
    "name": "5 Mbps",
    "value": 5120
  }
],

```

```

{
  "name": "6 Mbps",
  "value": 6144
},
{
  "name": "8 Mbps",
  "value": 8192
},
{
  "name": "10 Mbps",
  "value": 10240
},
{
  "name": "12 Mbps",
  "value": 12288
},
{
  "name": "16 Mbps",
  "value": 16384
},
{
  "name": "20 Mbps",
  "value": 20480
},
{
  "name": "24 Mbps",
  "value": 24576
},
{
  "name": "25 Mbps",
  "value": 25600
},
{
  "name": "26 Mbps",
  "value": 26624
},
{
  "name": "31 Mbps",
  "value": 31744
},
{
  "name": "32 Mbps",
  "value": 32768
},
{
  "name": "41 Mbps",
  "value": 41984
},
{
  "name": "50 Mbps",
  "value": 51200
},
{
  "name": "52 Mbps",
  "value": 53248
},
{
  "name": "62 Mbps",
  "value": 63488
}
],
"audio-streams": [
  {
    "name": "Audio stream 1",
    "value": 0
  },
  {
    "name": "Audio stream 2",
    "value": 1
  },
  {
    "name": "Audio stream 3",
    "value": 2
  }
]

```

```

    },
    {
      "name": "Audio stream 4",
      "value": 3
    }
  ],
  "audio-channels": [
    {
      "name": "Follow Input",
      "value": 0
    },
    {
      "name": "2 Channels",
      "value": 2
    },
    {
      "name": "4 Channels",
      "value": 4
    },
    {
      "name": "6 Channels",
      "value": 6
    },
    {
      "name": "8 Channels",
      "value": 8
    }
  ],
  "audio-chmap": [
    {
      "name": "Source channels 1/2",
      "value": 0
    },
    {
      "name": "Source channels 3/4",
      "value": 1
    },
    {
      "name": "Source channels 5/6",
      "value": 2
    },
    {
      "name": "Source channels 7/8",
      "value": 3
    }
  ],
  "audio-kbps": [
    {
      "name": " 16 Kbps",
      "value": 16
    },
    {
      "name": " 32 Kbps",
      "value": 32
    },
    {
      "name": " 48 Kbps",
      "value": 48
    },
    {
      "name": " 64 Kbps",
      "value": 64
    },
    {
      "name": " 96 Kbps",
      "value": 96
    },
    {
      "name": "128 Kbps",
      "value": 128
    },
    {
      "name": "192 Kbps",

```

```

    "value": 192
  },
  {
    "name": "256 Kbps",
    "value": 256
  }
],
"gop-sec": [
  {
    "name": " 1 sec",
    "value": 1
  },
  {
    "name": " 2 sec",
    "value": 2
  },
  {
    "name": " 5 sec",
    "value": 5
  },
  {
    "name": "10 sec",
    "value": 10
  },
  {
    "name": "30 sec",
    "value": 30
  },
  {
    "name": "60 sec",
    "value": 60
  }
],
"video-range": [
  {
    "name": "Full range (0-255)",
    "value": 1
  },
  {
    "name": "Limited range (16-235)",
    "value": 0
  }
],
"stat-sec": [
  {
    "name": " 1 sec",
    "value": 1
  },
  {
    "name": " 5 sec",
    "value": 5
  },
  {
    "name": "10 sec",
    "value": 10
  },
  {
    "name": "30 sec",
    "value": 30
  },
  {
    "name": "60 sec",
    "value": 60
  }
],
"video-codec": [
  {
    "name": "H.264",
    "value": 0
  },
  {
    "name": "H.265 (HEVC)",
    "value": 1
  }
]

```



```

    }
  ],
  "video-ar-convert-mode": [
    {
      "name": "Ignore",
      "value": 0
    },
    {
      "name": "Cropping",
      "value": 1
    },
    {
      "name": "Padding",
      "value": 2
    }
  ]
},
"rec": {
  "mode": [
    {
      "name": "Ordinary recording",
      "value": 0
    },
    {
      "name": "Loop recording",
      "value": 1
    }
  ],
  "file-prefix": [
    {
      "name": "Custom prefix",
      "value": 0
    },
    {
      "name": "Creation time",
      "value": 1
    }
  ],
  "file-suffix": [
    {
      "name": "mp4",
      "value": 0
    },
    {
      "name": "mov",
      "value": 1
    }
  ],
  "time-unit": [
    {
      "name": "5 minutes",
      "value": 5
    },
    {
      "name": "10 minutes",
      "value": 10
    },
    {
      "name": "30 minutes",
      "value": 30
    },
    {
      "name": "40 minutes",
      "value": 40
    },
    {
      "name": "50 minutes",
      "value": 50
    },
    {
      "name": "60 minutes",
      "value": 60
    }
  ],

```

```

    {
      "name": "90 minutes",
      "value": 90
    },
    {
      "name": "120 minutes",
      "value": 120
    }
  ]
},
"nas": {
  "proto": [
    {
      "name": "NFS - Network File System",
      "value": 0
    },
    {
      "name": "CIFS - Common Internet File System",
      "value": 1
    }
  ]
},
"color-range": {
  "contrast": {
    "max": 200,
    "min": 50,
    "def": 100
  },
  "brightness": {
    "max": 100,
    "min": -100,
    "def": 0
  },
  "saturation": {
    "max": 200,
    "min": 0,
    "def": 100
  },
  "hue": {
    "max": 90,
    "min": -90,
    "def": 0
  }
},
"video-format": {
  "input-color-fmt": [
    {
      "name": "RGB",
      "value": 1
    },
    {
      "name": "YUV BT.601",
      "value": 2
    },
    {
      "name": "YUV BT.709",
      "value": 3
    },
    {
      "name": "YUV BT.2020",
      "value": 4
    }
  ],
  "output-color-fmt": [
    {
      "name": "YUV BT.601",
      "value": 2
    },
    {
      "name": "YUV BT.709",
      "value": 3
    }
  ],
},

```

```

"quant-range": [
  {
    "name": "Full range (0-255)",
    "value": 1
  },
  {
    "name": "Limited range (16-235)",
    "value": 2
  }
],
"sat-range": [
  {
    "name": "Full range (0-255)",
    "value": 1
  },
  {
    "name": "Limited range (16-235)",
    "value": 2
  },
  {
    "name": "Extended GAMUT range (1-254)",
    "value": 3
  }
]
},
"living": {
  "max-bandwidth": 16384,
  "ttl": {
    "max": 255,
    "min": 0,
    "def": 0
  },
  "conn-timeout": {
    "max": 30000,
    "min": 1000
  },
  "retry-duration": {
    "max": 10000,
    "min": 0
  },
  "rtmp": {
    "def-conn-timeout": 10000,
    "def-retry-duration": 10000
  },
  "srt": {
    "def-conn-timeout": 3000,
    "def-retry-duration": 10000,
    "latency": {
      "max": 8000,
      "min": 30,
      "def": 120
    },
    "bandwidth": {
      "max": 100,
      "min": 5,
      "def": 25
    },
  },
  "aes": [
    {
      "name": "Not Used",
      "value": 0
    },
    {
      "name": "AES-128",
      "value": 16
    },
    {
      "name": "AES-192",
      "value": 24
    },
    {
      "name": "AES-256",
      "value": 32
    }
  ]
}

```

```

    }
  ],
  "mtu": {
    "max": 1500,
    "min": 232,
    "def": 1496
  }
},
"ndi": {
  "transport-mode": [
    {
      "name": "UDP (Unicast)",
      "value": 0
    },
    {
      "name": "UDP (Multicast)",
      "value": 1
    },
    {
      "name": "RUDP (Unicast)",
      "value": 2
    },
    {
      "name": "TCP (Uni-Connection)",
      "value": 3
    },
    {
      "name": "TCP (Multi-Connection)",
      "value": 4
    }
  ]
},
"udp": {
  "mtu": {
    "max": 1500,
    "min": 228,
    "def": 1496
  }
}
},
"surface": {
  "image": {
    "max-count": 8,
    "max-width": 1920,
    "max-height": 1080,
    "max-size-kb": 512
  },
  "type": [
    {
      "name": "Text",
      "value": 0
    },
    {
      "name": "System Time",
      "value": 1
    },
    {
      "name": "Image",
      "value": 2
    }
  ]
},
"image-type": [
  {
    "name": "JPEG",
    "value": 0
  },
  {
    "name": "PNG",
    "value": 1
  }
],
"condition": [
  {

```

```

    "name": "Always",
    "value": 0
  }
],
"position": [
  {
    "name": "Custom",
    "value": 0
  },
  {
    "name": "Left Top Corner",
    "value": 1
  },
  {
    "name": "Top Right Corner",
    "value": 2
  },
  {
    "name": "Top Center",
    "value": 5
  },
  {
    "name": "Left Bottom Corner",
    "value": 3
  },
  {
    "name": "Bottom Right Corner",
    "value": 4
  },
  {
    "name": "Bottom Center",
    "value": 6
  },
  {
    "name": "Center",
    "value": 7
  }
],
"font-family": [
  {
    "name": "Source Han Sans",
    "value": 0
  },
  {
    "name": "Lato",
    "value": 1
  }
],
"font-style": [
  {
    "name": "Normal",
    "value": 0
  },
  {
    "name": "Italic",
    "value": 1
  },
  {
    "name": "Oblique",
    "value": 2
  }
],
"font-weight": [
  {
    "name": "Normal",
    "value": 0
  },
  {
    "name": "Bold",
    "value": 1
  }
],
"time-format": [

```

```
{
  "name": "YYYY-MM-DD hh:mm:ss",
  "value": 0
},
{
  "name": "MM/DD/YYYY hh:mm:ss",
  "value": 1
},
{
  "name": "DD/MM/YYYY hh:mm:ss",
  "value": 2
},
{
  "name": "YYYY-MM-DD",
  "value": 3
},
{
  "name": "MM/DD/YYYY",
  "value": 4
},
{
  "name": "DD/MM/YYYY",
  "value": 5
},
{
  "name": "hh:mm:ss",
  "value": 6
},
{
  "name": "hh:mm",
  "value": 7
}
]
}
```

get-status 接口

实时获取设备的工作状态，主要有直播状态、固件升级状态等。

请求方式

```
GET http://ip/usapi?method=get-status
```

参数	说明
method	方法名称: get-status

返回数据

JSON 结构如下:

```
{
  "result": 0, // 返回状态
  "cur-status": 64, // 设备运行状态掩码
  "cur-time": "2021-01-11 ...", // 设备当前时间
  "box-name": "", // 设备名称
  "input-source": 1, // 输入信号源
  "input-device": 1, // 输入设备
  "cpu-temperature": 61100,
  "enable-ndi-hx3": 0,
  "codec": { // 编码器状态
    "main-stream": {},
    "sub-stream": {},
    "audio": {}
  },
  "sysstat": {}, // 设备运行状态
  "live-status": { // 直播状态
    "live": []
  },
  "upgrade-status": {}, // 固件升级过程状态
  "rec-status": { // 录制状态
    "rec": []
  },
  "format-status": {}, // 磁盘格式化过程状态
  "disk-test": {}, // 磁盘性能测试过程状态
  "nas": {}, // 网络存储连接过程状态
  "living-test": {}, // 直播推流测试过程状态
  "check-upgrade": {}, // 固件在线检测过程状态
  "conn-wifi": {}, // 无线网络连接过程状态
  "input-signal": {}, // 输入信号信息
  "disk-info": {}, // 磁盘信息
  "wifi": {}, // 无线网信息
  "softap": {}, // AP 网络信息
  "eth": {}, // 以太网信息
  "mobile": {}, // 移动网络信息
  "rndis": {}, // USB 网络信息
  "upgrade": {}, // 新版本固件信息
  "channel-count": 2,
  "vumeters": [
    33,
    32
  ]
}
```

返回状态

```
"result": 0
```

设备运行状态掩码

```
"cur-status": 65552
```

设备当前运行的各种状态，通过不同掩码值表示，各个状态对应的掩码值请参考 [设备状态掩码](#)。采用以下计算方式来判断：

- 1、通设备状态掩码表可知，正在进行固件升级的值为：statusUpgrade = 0x4000
- 2、如果 `cur-status & statusUpgrade = statusUpgrade`，则说明设备正处于固件升级的状态中

输入信号状态

```
"input-signal": {
  "status": 0,
  "cx": 0,
  "cy": 0,
  "interlaced": 0,
  "frame-rate": 0.00,
  "channel-valid": 0,
  "is-lpcm": 0,
  "bits-per-sample": 0,
  "sample-rate": 0,
  "hdmi": {
    "status": 0,
    "cx": 0,
    "cy": 0,
    "interlaced": 0,
    "interlaced-scale": 1,
    "frame-rate": 0.00,
    "channel-valid": 0,
    "is-lpcm": 0,
    "bits-per-sample": 0,
    "sample-rate": 0
  },
  "sdi": {
    "status": 0,
    "cx": 0,
    "cy": 0,
    "interlaced": 0,
    "interlaced-scale": 1,
    "frame-rate": 0.00,
    "channel-valid": 0,
    "is-lpcm": 0,
    "bits-per-sample": 0,
    "sample-rate": 0
  }
}
```

直播状态

```
"live-status": {
  [
    {
      "id": 0,
      "type": 130,
      "is-use": 1,
      "is-skd-running": 0,
      "name": "NDI HX",
      "run-ms": 231465213,
      "result": 22,
      "main-inst-bps": 993141,
      "sub-inst-bps": 103444,
    }
  ]
}
```

直播推流测试过程状态

```
"living-test": {
  "upload-bps": 0,
  "percent": 0,
  "result": 27,
  "net": 0,
  "client-id": ""
}
```



```
}
```

录制状态

```
"rec-status": {  
  "rec": [  
    {  
      "id": 1,  
      "type": 1,  
      "is-use": 1,  
      "is-skd-runnung": 0,  
      "result": 2,  
      "run-ms": 1700,  
      "parted-num": 1,  
      "video-frame-count": 70,  
      "audio-frame-count": 54  
    }  
  ]  
}
```

磁盘格式化过程状态

```
"format-status": {  
  "type": 0,  
  "percent": 0,  
  "result": 27,  
  "client-id": ""  
}
```

磁盘测试过程状态

```
"disk-test": {  
  "type": 0,  
  "read-bps": 0,  
  "write-bps": 0,  
  "percent": 0,  
  "result": 27,  
  "client-id": ""  
}
```

网络存储连接过程状态

```
"nas": [  
  {  
    "id": 0,  
    "type": 2,  
    "result": 22  
  }  
]
```

在线固件检测过程状态

```
"check-upgrade": {  
  "result": 0,  
  "client-id": ""  
}
```

新版本固件信息

```
"upgrade": {  
  "ver": "",  
  "date": "",  
  "size-byte": 0,  
  "info": []  
}
```

固件升级过程状态

```
"upgrade-status": {  
  "step": 0,  
  "percent": 0,  
  "result": 27,  
  "client-id": "",  
  "mode": "none"  
}
```

无线网信息

```
"wifi": {  
  "name": "MWL1",  
  "level": 0,  
  "ip": "192.168.8.249",  
  "mask": "255.255.255.0",  
  "router": "192.168.8.1",  
  "dns": "192.168.8.1"  
}
```

以太网信息

```
"eth": {  
  "ip": "10.10.107.212",  
  "mask": "255.255.0.0",  
  "router": "10.10.0.1",  
  "dns": "10.0.0.3"  
}
```

移动网络信息

```
"mobile": {  
  "ip": "",  
  "mask": "",  
  "router": "",  
  "dns": ""  
}
```

get-settings 接口

获取设备的设置信息。

请求方式

```
GET http://ip/usapi?method=get-settings
```

参数	说明
method	方法名称: get-settings

返回数据

JSON 结构如下:

```
{
  "result": 0, // 返回状态码
  "name": "Ultra...", // 设备名称
  "is-check-update": 1, // 是否允许在线检测固件版本
  "audio-sync-offset": 0, // 音频时延 (毫秒)
  "udp-mtu": 1496, // UDP协议 最大传输单元
  "enable-ndi-hx3": 0, // 是否启用 NDI|HX3
  "softap": {}, // AP 配置信息
  "date-time": {}, // 时间日期信息
  "input-source": {}, // 输入源信息
  "video-color": {}, // 视频相关信息
  "volume": {}, // 音频相关信息
  "enable-deinterlace": 1, // 是否进行视频去隔行
  "main-stream": {}, // 主码流配置信息
  "sub-stream": {}, // 次码流配置信息
  "audio": {}, // 音频配置信息
  "audio-streams": [...], // 音频码流列表
  "eth": {}, // 以太网信息
  "wifi": {}, // Wi-Fi 信息
  "rndis": {}, // USB 网络信息
  "stream-server": [...], // 直播服务器列表
  "video-input-format": {}, // 视频输入格式信息
  "video-output-format": {}, // 视频输出格式信息
  "use-nosignal-file": 1, // 是否启用无信号图片
  "nosignal-files": [...], // 无信号图片列表
  "nas": [...], // 网络存储服务器列表
}
```

返回示例:

```
{
  "result": 0,
  "name": "Ultra Encode B313220727002",
  "is-check-update": 1,
  "living-kbps": 32768,
  "audio-sync-offset": 0,
  "enable-advanced-pcr": 0,
  "udp-mtu": 1496,
  "enable-ndi-hx3": 0,
  "softap": {
    "is-softap": 1,
    "is-visible": 1,
    "softap-ssid": "B313220727002",
    "softap-passwd": "20727002"
  },
  "date-time": {
    "timezone": "Asia/Shanghai",
    "is-auto": 1,
    "ntp-server": "0.pool.ntp.org",
    "ntp-server-backup": "1.pool.ntp.org"
  }
}
```

```

},
"input-source": {
  "source": 1,
  "mixer": {
    "input-device": 2,
    "is-hdmi-top": 0,
    "type": 0,
    "location": 2
  }
},
"video-color": {
  "hdmi": {
    "contrast": 100,
    "brightness": 0,
    "saturation": 100,
    "hue": 0
  },
  "sdi": {
    "contrast": 100,
    "brightness": 0,
    "saturation": 100,
    "hue": 0
  }
},
"volume": {
  "is-spi": 1,
  "spi-gain": 0,
  "is-linein": 1,
  "linein-gain": 127,
  "is-lineout": 1,
  "lineout-gain": 0,
  "enable-mic-bias": 0
},
"enable-deinterlace": 1,
"main-stream": {
  "is-auto": 0,
  "codec": 1,
  "cx": 1920,
  "cy": 1080,
  "duration": 0,
  "kbps": 8192,
  "gop": 60,
  "fourcc": 0,
  "profile": 0,
  "cbrstat": 60,
  "fullrange": 0,
  "is-vbr": 0,
  "min-vbr-qp": 0,
  "max-vbr-qp": 0,
  "is-time-code-sei": 1,
  "is-closed-caption-sei": 0,
  "ar-convert-mode": 2,
  "rotation": 0,
  "mirroring": 0
},
"sub-stream": {
  "enable": 1,
  "codec": 1,
  "cx": 1920,
  "cy": 1080,
  "duration": 333333,
  "kbps": 4096,
  "gop": 60,
  "fourcc": 0,
  "profile": 0,
  "cbrstat": 60,
  "fullrange": 0,
  "is-vbr": 0,
  "min-vbr-qp": 0,
  "max-vbr-qp": 0,
  "is-time-code-sei": 1,
  "is-closed-caption-sei": 0,
  "ar-convert-mode": 2,

```

```

    "rotation": 0,
    "mirroring": 0
  },
  "audio": {
    "sample-rate": 48000,
    "channels": 0,
    "kbps": 256,
    "ch0": 0,
    "ch1": 1,
    "ch2": 1,
    "ch3": 3
  },
  "audio-streams": [
    {
      "sample-rate": 48000,
      "channels": 0,
      "kbps": 256,
      "ch0": 0,
      "ch1": 1,
      "ch2": 1,
      "ch3": 3
    },
    {
      "sample-rate": 48000,
      "channels": 2,
      "kbps": 32,
      "ch0": 2,
      "ch1": 1,
      "ch2": 2,
      "ch3": 3
    },
    {
      "sample-rate": 48000,
      "channels": 2,
      "kbps": 128,
      "ch0": 3,
      "ch1": 1,
      "ch2": 2,
      "ch3": 3
    },
    {
      "sample-rate": 48000,
      "channels": 2,
      "kbps": 192,
      "ch0": 3,
      "ch1": 1,
      "ch2": 2,
      "ch3": 3
    }
  ],
  "eth": {
    "is-dhcp": 1,
    "ip": "",
    "mask": "",
    "router": "",
    "dns": ""
  },
  "wifi": {
    "is-dhcp": 1,
    "ip": "",
    "mask": "",
    "router": "",
    "dns": ""
  },
  "rndis": {
    "ip": "192.168.66.1",
    "mask": "255.255.255.0"
  },
  "stream-server": [
    {
      "id": 0,
      "type": 130,
      "name": "NDI HX",

```

```

    "is-use": 0,
    "source-name": "#serial-no#",
    "group-name": "public",
    "enable-discovery": 0,
    "discovery-server": "",
    "transport-mode": 3,
    "mcast-addr": "",
    "mcast-mask": "",
    "mcast-ttl": 0,
    "enable-fail-over": 0,
    "fail-over-ndi-name": "",
    "fail-over-ip-addr": "",
    "enable-web-control": 0,
    "enable-ptz-control": 0,
    "main-stream": 0,
    "prvw-stream": 1,
    "audio": 2,
    "opt": 0,
    "is-media-hub": 0
  }
],
"video-input-format": {
  "hdmi": {
    "is-color-fmt": 0,
    "color-fmt": 1,
    "is-quant-range": 0,
    "quant-range": 1
  },
  "sdi": {
    "is-color-fmt": 0,
    "color-fmt": 1,
    "is-quant-range": 0,
    "quant-range": 1
  }
},
"video-output-format": {
  "hdmi": {
    "is-color-fmt": 0,
    "color-fmt": 3,
    "is-quant-range": 0,
    "quant-range": 2,
    "is-sat-range": 0,
    "sat-range": 2
  },
  "sdi": {
    "is-color-fmt": 0,
    "color-fmt": 3,
    "is-quant-range": 0,
    "quant-range": 2,
    "is-sat-range": 0,
    "sat-range": 2
  }
},
"use-nosignal-file": 1,
"nosignal-files": [
  {
    "id": 0,
    "is-use": 0,
    "is-edit": 0,
    "file-path": "/no-signal/default0.jpg",
    "time": 0
  },
  {
    "id": 1,
    "is-use": 1,
    "is-edit": 0,
    "file-path": "/no-signal/default1.jpg",
    "time": 0
  },
  {
    "id": 2,
    "is-use": 0,
    "is-edit": 1,

```

```
    "file-path": "/no-signal/default2.jpg",
    "time": 16687566986732126
  }
],
"nas": [
  {
    "id": 0,
    "type": 2,
    "proto": 0,
    "is-use": 1,
    "addr": "10.10.14.193",
    "mount": "/home/magewell/nfs"
  }
]
}
```

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 http://ip/usapi?method=ping
```

参数	说明
method	ping

返回数据

```
{
  "result": 0,
  "micro-sec": 149564858458,
  "cur-status": 65552
}
```

属性	说明
result	0: 设备可以访问。返回其它值请参考 API 状态码 。
cur-status	设备当前工作状态掩码值。相关掩码值请参考 设备状态掩码 。

get-report 接口

获取设备信息、状态和配置的相关信息。

请求方式

```
GET http://ip/usapi?method=get-report
```

参数	说明
method	方法名称: get-report

返回数据

```
{  
  <div class="report-summary">  
    ...  
    <h2>DEVICE</h2>  
    ...  
    <h2>STATUS</h2>  
    ...  
    <h2>SETTINGS</h2>  
    ...  
  </div>  
}
```

export-report 接口

将设备信息、状态和配置等相关信息以 html 文件保存到本地。

请求方式

```
GET http://ip/usapi?method=export-report&file-name=...
```

参数	说明
method	方法名称: export-report
file-name	保存的文件路径

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

get-logs 接口

获取系统日志列表，系统最多记录最近 1000 条数据，仅管理员有权限。

请求方式

```
GET http://ip/usapi?method=get-logs&types=all
```

参数	说明
method	方法名称: get-logs
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。

返回数据

```
{
  "result": 0
  "logs": [
    {
      "type": "warn",
      "time": "2022-03-19 09:53:03.047",
      "message": "USB state: disconnected"
    },
    ...
  ]
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

export-logs 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

```
GET http://ip/usapi?method=export-logs&file-name=...
```

参数	说明
method	方法名称: export-logs
file-name	保存的文件路径

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

clear-logs 接口

清除全部系统日志，仅管理员有权限。

请求方式

```
GET http://ip/usapi?method=clear-logs
```

参数	说明
method	方法名称: clear-logs

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

set-name 接口

设置设备名称。

请求方式

```
GET http://ip/usapi?method=set-name&name=xxx
```

参数	说明
method	方法名称: set-name
name	设备名称, 字符要求: 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 . _ + '[]() 组成, 并且不能以空格开头或结束

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-first-over 接口

首次访问设备时，需要调用该接口进行初始化设置，主要是设置 **设备名称**。

通过 [get-status](#) 接口可以获取到以下信息：

```
{
  "cur-status": 65552           // 设备运行状态掩码
  ...
}
```

设备首次运行的值为：[statusFirst\(0x01\)](#)，如果 `cur-status & statusFirst = statusFirst`，那么说明设备是首次运行

请求方式

```
GET http://ip/usapi?method=set-first-over&name=xxx
```

参数	说明
method	方法名称：set-first-over
name	设备名称，字符要求： 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 . _ - + []() 组成，并且不能以空格开头或结束

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态，0：操作成功，返回其它值请参考 API 状态码

set-softap 接口

修改 AP 配置。

请求方式

```
GET http://ip/usapi?method=set-softap&is-softap=0&is-visible=1&softap-ssid=xxx&softap-passwd=xxx
```

参数	说明
method	方法名称: set-softap
is-softap	保留, 默认值: 1
is-visible	保留, 默认值: 1
softap-ssid	保留, 默认值: 产品序列号
softap-passwd	SSID 密码, 明文存储, 字符要求: 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 ._-+'[]() 组成, 并且不能以空格开头或结束

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-ssdp 接口

是否开启 UPNP。

请求方式

```
GET http://ip/usapi?method=set-ssdp&is-ssdp=1
```

参数	说明
method	方法名称: set-ssdp
is-ssdp	是否启用 0: 关闭 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-volume 接口

调节输入信号、耳机、麦克风音量的增益。

通过 [get-info](#) 可以获取到录制相关参数的取值范围：

```
"audio-range": {  
  "spi": [],      // 输入信号音量增益取值范围  
  "linein": [],   // Line In 音量增益取值范围  
  "lineout": [],  // Line Out 音量增益取值范围  
}
```

请求方式

```
GET http://ip/usapi?method=set-volume&is-linein=1&linein-gain=0&is-spi=1&spi-gain=0&is-lineout=1&lineout-gain=0
```

参数	说明
method	方法名称: set-volume
is-linein	设置麦克风音量增益启用状态, 0: 静音 1: 启用
linein-gain	设置麦克风音量增益, 默认值: 0
is-spi	设置输入信号音量增益启用状态, 0: 静音 1: 启用
spi-gain	设置输入信号音量增益, 默认值: 0
is-lineout	设置耳机音量增益启用状态, 0: 静音 1: 启用
lineout-gain	设置耳机音量增益, 默认值: 0

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-sync-offset 接口

设置音频时延时间，单位毫秒。音频滞后时，建议设置为负值；音频超前时，建议设置为正值。

请求方式

```
GET http://ip/usapi?method=set-sync-offset&audio-sync-offset=100
```

参数	说明
method	方法名称: set-sync-offset
audio-sync-offset	设置音频时延时间，取值范围为[-200,200]毫秒

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态，0：操作成功，返回其它值请参考 API 状态码

add-nosignal-file 接口

添加无输入信号时显示的图片，仅支持分辨率小于 1920x1080，文件大小小于 1MB 的 JPEG 图片，最多支持添加 2 张。

请求方式

```
POST http://ip/usapi?method=add-nosignal-file
```

参数	说明
method	方法名称: add-nosignal-file

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

del-nosignal-file 接口

删除用户添加的无信号图片，默认图片不可删除。

请求方式

```
GET http://ip/usapi?method=del-nosignal-file&id=0
```

参数	说明
method	方法名称: del-nosignal-file
id	图片 ID

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-nosignal-file 接口

当启用无信号图片且设备无输入信号时，显示选中的图片。

请求方式

```
GET http://ip/usapi?method=set-nosignal-file&id=0
```

参数	说明
method	方法名称: set-nosignal-file
id	图片 ID

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

use-nosignal-file 接口

是否启用无信号图片。当设备无输入信号时，显示用户设备的图片。

请求方式

```
GET http://ip/usapi?method=use-nosignal-file&use-nosignal-file=0
```

参数	说明
method	方法名称：use-nosignal-file
use-nosignal-file	是否启用无信号图片 0: 不启用 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态，0：操作成功，返回其它值请参考 API 状态码

set-time-zone 接口

设置时区。

请求方式

```
GET http://ip/usapi?method=set-time-zone&timezone=Asia/Shanghai
```

参数	说明
method	set-time-zone
timezone	时区，如：Asia/Shanghai，具体值请参考下文常见时区

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态，0：操作成功，返回其它值请参考 API 状态码

常见时区

```
[  
  'Africa/Abidjan',  
  'Africa/Accra',  
  'Africa/Addis_Ababa',  
  'Africa/Algiers',  
  'Africa/Asmara',  
  'Africa/Bamako',  
  'Africa/Bangui',  
  'Africa/Banjul',  
  'Africa/Bissau',  
  'Africa/Blantyre',  
  'Africa/Brazzaville',  
  'Africa/Bujumbura',  
  'Africa/Cairo',  
  'Africa/Casablanca',  
  'Africa/Ceuta',  
  'Africa/Conakry',  
  'Africa/Dakar',  
  'Africa/Dar_es_Salaam',  
  'Africa/Djibouti',  
  'Africa/Douala',  
  'Africa/El_Aaiun',  
  'Africa/Freetown',  
  'Africa/Gaborone',  
  'Africa/Harare',  
  'Africa/Johannesburg',  
  'Africa/Juba',  
  'Africa/Kampala',  
  'Africa/Khartoum',  
  'Africa/Kigali',  
  'Africa/Kinshasa',  
  'Africa/Lagos',  
  'Africa/Libreville',  
  'Africa/Lome',  
  'Africa/Luanda',  
  'Africa/Lubumbashi',  
  'Africa/Lusaka',  
  'Africa/Malabo',  
  'Africa/Maputo',  
  'Africa/Maseru',
```


'Africa/Mbabane',
'Africa/Mogadishu',
'Africa/Monrovia',
'Africa/Nairobi',
'Africa/Ndjamena',
'Africa/Niamey',
'Africa/Nouakchott',
'Africa/Ouagadougou',
'Africa/Porto-Novo',
'Africa/Sao_Tome',
'Africa/Timbuktu',
'Africa/Tripoli',
'Africa/Tunis',
'Africa/Windhoek',
'America/Adak',
'America/Anchorage',
'America/Anguilla',
'America/Antigua',
'America/Araguaina',
'America/Argentina/Buenos_Aires',
'America/Argentina/Catamarca',
'America/Argentina/ComodRivadavia',
'America/Argentina/Cordoba',
'America/Argentina/Jujuy',
'America/Argentina/La_Rioja',
'America/Argentina/Mendoza',
'America/Argentina/Rio_Gallegos',
'America/Argentina/Salta',
'America/Argentina/San_Juan',
'America/Argentina/San_Luis',
'America/Argentina/Tucuman',
'America/Argentina/Ushuaia',
'America/Aruba',
'America/Asuncion',
'America/Atikokan',
'America/Atka',
'America/Bahia',
'America/Bahia_Banderas',
'America/Barbados',
'America/Belem',
'America/Belize',
'America/Blanc-Sablon',
'America/Boa_Vista',
'America/Bogota',
'America/Boise',
'America/Buenos_Aires',
'America/Cambridge_Bay',
'America/Campo_Grande',
'America/Cancun',
'America/Caracas',
'America/Cayenne',
'America/Cayman',
'America/Chicago',
'America/Chihuahua',
'America/Costa_Rica',
'America/Creston',
'America/Cuiaba',
'America/Curacao',
'America/Danmarkshavn',
'America/Dawson',
'America/Dawson_Creek',
'America/Denver',
'America/Detroit',
'America/Dominica',
'America/Edmonton',
'America/Eirunepe',
'America/El_Salvador',
'America/Ensenada',
'America/Fort_Nelson',
'America/Fort_Wayne',
'America/Fortaleza',
'America/Glace_Bay',
'America/Godthab',

'America/Goose_Bay',
'America/Grand_Turk',
'America/Grenada',
'America/Guadeloupe',
'America/Guatemala',
'America/Guayaquil',
'America/Guyana',
'America/Halifax',
'America/Havana',
'America/Hermosillo',
'America/Indiana/Indianapolis',
'America/Indiana/Knox',
'America/Indiana/Marengo',
'America/Indiana/Petersburg',
'America/Indiana/Tell_City',
'America/Indiana/Vevay',
'America/Indiana/Vincennes',
'America/Indiana/Winamac',
'America/Indianapolis',
'America/Inuvik',
'America/Iqaluit',
'America/Jamaica',
'America/Juneau',
'America/Kentucky/Louisville',
'America/Kentucky/Monticello',
'America/Kralendijk',
'America/La_Paz',
'America/Lima',
'America/Los_Angeles',
'America/Louisville',
'America/Lower_Princes',
'America/Maceio',
'America/Managua',
'America/Manaus',
'America/Marigot',
'America/Martinique',
'America/Matamoros',
'America/Mazatlan',
'America/Mendoza',
'America/Menominee',
'America/Merida',
'America/Metlakatla',
'America/Mexico_City',
'America/Miquelon',
'America/Moncton',
'America/Monterrey',
'America/Montevideo',
'America/Montreal',
'America/Montserrat',
'America/Nassau',
'America/New_York',
'America/Nipigon',
'America/Nome',
'America/Noronha Atlantic islands',
'America/North_Dakota/Beulah',
'America/North_Dakota/Center',
'America/North_Dakota/New_Salem',
'America/Ojinaga',
'America/Panama',
'America/Pangnirtung',
'America/Paramaribo',
'America/Phoenix',
'America/Port-au-Prince',
'America/Port_of_Spain',
'America/Porto_Acre',
'America/Porto_Velho',
'America/Puerto_Rico',
'America/Punta_Arenas',
'America/Rainy_River',
'America/Rankin_Inlet',
'America/Recife',
'America/Regina',
'America/Resolute',

'America/Rio_Branco',
'America/Rosario',
'America/Santa_Isabel',
'America/Santarem',
'America/Santiago',
'America/Santo_Domingo',
'America/Sao_Paulo',
'America/Scoresbysund',
'America/Shiprock',
'America/Sitka',
'America/St_Barthelemy',
'America/St_Johns',
'America/St_Kitts',
'America/St_Lucia',
'America/St_Thomas',
'America/St_Vincent',
'America/Swift_Current',
'America/Tegucigalpa',
'America/Thule',
'America/Thunder_Bay',
'America/Tijuana',
'America/Toronto',
'America/Tortola',
'America/Vancouver',
'America/Virgin',
'America/Whitehorse',
'America/Winnipeg',
'America/Yakutat',
'America/Yellowknife',
'Asia/Aden',
'Asia/Almaty',
'Asia/Amman',
'Asia/Anadyr',
'Asia/Aqtau',
'Asia/Aqtobe',
'Asia/Ashgabat',
'Asia/Ashkhabad',
'Asia/Atyrau',
'Asia/Baghdad',
'Asia/Bahrain',
'Asia/Baku',
'Asia/Bangkok',
'Asia/Barnaul',
'Asia/Beirut',
'Asia/Bishkek',
'Asia/Brunei',
'Asia/Calcutta',
'Asia/Chita',
'Asia/Choibalsan',
'Asia/Chongqing',
'Asia/Chungking',
'Asia/Colombo',
'Asia/Dacca',
'Asia/Damascus',
'Asia/Dhaka',
'Asia/Dili',
'Asia/Dubai',
'Asia/Dushanbe',
'Asia/Famagusta',
'Asia/Gaza',
'Asia/Harbin',
'Asia/Hebron West',
'Asia/Ho_Chi_Minh',
'Asia/Hong_Kong',
'Asia/Hovd',
'Asia/Irkutsk',
'Asia/Istanbul',
'Asia/Jakarta',
'Asia/Jayapura',
'Asia/Jerusalem',
'Asia/Kabul',
'Asia/Kamchatka',
'Asia/Karachi',

'Asia/Kashgar',
'Asia/Kathmandu',
'Asia/Katmandu',
'Asia/Khandyga',
'Asia/Kolkata',
'Asia/Krasnoyarsk',
'Asia/Kuala_Lumpur',
'Asia/Kuching',
'Asia/Kuwait',
'Asia/Macao',
'Asia/Macau',
'Asia/Magadan',
'Asia/Makassar',
'Asia/Manila',
'Asia/Muscat',
'Asia/Nicosia',
'Asia/Novokuznetsk',
'Asia/Novosibirsk',
'Asia/Omsk',
'Asia/Oral',
'Asia/Phnom_Penh',
'Asia/Pontianak',
'Asia/Pyongyang',
'Asia/Qatar',
'Asia/Qyzylorda',
'Asia/Rangoon',
'Asia/Riyadh',
'Asia/Saigon',
'Asia/Sakhalin',
'Asia/Samarkand',
'Asia/Seoul',
'Asia/Shanghai',
'Asia/Singapore',
'Asia/Srednekolymsk',
'Asia/Taipei',
'Asia/Tashkent',
'Asia/Tbilisi',
'Asia/Tehran',
'Asia/Thimbu',
'Asia/Thimphu',
'Asia/Tokyo',
'Asia/Tomsk',
'Asia/Ujung_Pandang',
'Asia/Ulaanbaatar',
'Asia/Ulan_Bator',
'Asia/Urumqi',
'Asia/Ust-Nera',
'Asia/Vientiane',
'Asia/Vladivostok',
'Asia/Yangon',
'Asia/Yakutsk',
'Asia/Yekaterinburg',
'Asia/Yerevan',
'Atlantic/Azores',
'Atlantic/Bermuda',
'Atlantic/Canary',
'Atlantic/Cape_Verde',
'Atlantic/Faeroe',
'Atlantic/Faroe',
'Atlantic/Jan_Mayen',
'Atlantic/Madeira',
'Atlantic/Reykjavik',
'Atlantic/South_Georgia',
'Atlantic/St_Helena',
'Atlantic/Stanley',
'Australia/ACT',
'Australia/Adelaide',
'Australia/Brisbane',
'Australia/Broken_Hill',
'Australia/Canberra',
'Australia/Currie',
'Australia/Darwin',
'Australia/Eucla',

'Australia/Hobart',
'Australia/LHI',
'Australia/Lindeman',
'Australia/Lord_Howe',
'Australia/Melbourne',
'Australia/NSW',
'Australia/North',
'Australia/Perth',
'Australia/Queensland',
'Australia/South',
'Australia/Sydney',
'Australia/Tasmania',
'Australia/Victoria',
'Australia/West',
'Australia/Yancowinna',
'Europe/Amsterdam',
'Europe/Andorra',
'Europe/Astrakhan',
'Europe/Athens',
'Europe/Belfast',
'Europe/Belgrade',
'Europe/Berlin',
'Europe/Bratislava',
'Europe/Brussels',
'Europe/Bucharest',
'Europe/Budapest',
'Europe/Busingen',
'Europe/Chisinau',
'Europe/Copenhagen',
'Europe/Dublin',
'Europe/Gibraltar',
'Europe/Guernsey',
'Europe/Helsinki',
'Europe/Isle_of_Man',
'Europe/Istanbul',
'Europe/Jersey',
'Europe/Kaliningrad',
'Europe/Kiev',
'Europe/Kirov',
'Europe/Lisbon',
'Europe/Ljubljana',
'Europe/London',
'Europe/Luxembourg',
'Europe/Madrid',
'Europe/Malta',
'Europe/Mariehamn',
'Europe/Minsk',
'Europe/Monaco',
'Europe/Moscow',
'Europe/Nicosia',
'Europe/Oslo',
'Europe/Paris',
'Europe/Podgorica',
'Europe/Prague',
'Europe/Riga',
'Europe/Rome',
'Europe/Samara',
'Europe/San_Marino',
'Europe/Sarajevo',
'Europe/Saratov',
'Europe/Simferopol',
'Europe/Skopje',
'Europe/Sofia',
'Europe/Stockholm',
'Europe/Tallinn',
'Europe/Tirane',
'Europe/Tiraspol',
'Europe/Ulyanovsk',
'Europe/Uzhgorod',
'Europe/Vaduz',
'Europe/Vatican',
'Europe/Vienna',
'Europe/Vilnius',

```
'Europe/Volgograd',
'Europe/Warsaw',
'Europe/Zagreb',
'Europe/Zaporozhye',
'Europe/Zurich',
'Indian/Antananarivo',
'Indian/Chagos',
'Indian/Christmas',
'Indian/Cocos',
'Indian/Comoro',
'Indian/Kerguelen',
'Indian/Mahe',
'Indian/Maldives',
'Indian/Mauritius',
'Indian/Mayotte',
'Indian/Reunion',
'Pacific/Apia',
'Pacific/Auckland',
'Pacific/Bougainville',
'Pacific/Chatham',
'Pacific/Chuuk',
'Pacific/Easter',
'Pacific/Efate',
'Pacific/Enderbury',
'Pacific/Fakaofu',
'Pacific/Fiji',
'Pacific/Funafuti',
'Pacific/Galapagos',
'Pacific/Gambier',
'Pacific/Guadalcanal',
'Pacific/Guam',
'Pacific/Honolulu',
'Pacific/Johnston',
'Pacific/Kiritimati',
'Pacific/Kosrae',
'Pacific/Kwajalein',
'Pacific/Majuro',
'Pacific/Marquesas',
'Pacific/Midway',
'Pacific/Nauru',
'Pacific/Niue',
'Pacific/Norfolk',
'Pacific/Noumea',
'Pacific/Pago_Pago',
'Pacific/Palau',
'Pacific/Pitcairn',
'Pacific/Pohnpei',
'Pacific/Ponape',
'Pacific/Port_Moresby',
'Pacific/Rarotonga',
'Pacific/Saipan',
'Pacific/Samoa',
'Pacific/Tahiti',
'Pacific/Tarawa',
'Pacific/Tongatapu',
'Pacific/Truk',
'Pacific/Wake',
'Pacific/Wallis',
'Pacific/Yap',
'UTC'
```

]

set-date-time 接口

设置时间和时间同步服务器。

请求方式

```
GET http://ip/usapi?method=set-date-time&is-auto=1&time=2019-09-10+15:00:00&...
```

参数	说明
method	方法名称: set-date-time
is-auto	是否自动获取时间, 0: 手动设置, 1: 根据时区自动获取
time	手动设置时间, 字符串类型, 格式如: 2019-09-10 15:10:00 当 is-auto=1 时, NTP 服务器必填, time="" 即可
ntp-server	NTP 服务器
ntp-server-backup	备用 NTP 服务器

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-udp-mtu 接口

设置 UDP 传输协议最大传输单元，用于 TS over UDP 和 TS over RTP 直播服务。

请求方式

```
GET http://ip/usapi?method=set-udp-mtu&udp-mtu=1496
```

参数	说明
method	方法名称: set-udp-mtu
udp-mtu	最大传输单元, 有效值:228 ~ 1500

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-analog-mic-bias 接口

是否为 Line In 启用偏置电压。

请求方式

```
GET http://ip/usapi?method=set-analog-mic-bias&enable-mic-bias=1
```

参数	说明
method	方法名称: set-analog-mic-bias
enable-mic-bias	启用状态 0: 禁用 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

set-video-mixer-config 接口

输入信号源设置合成模式时，配置 SDI 与 HDMI 输入合成参数。

通过 [get-info](#) 可以获取到合成相关参数的取值范围：

```
"input-source": {
  "video-mixer": {
    "types": [],           // 位置信息取值范围
    "pip": [],           // 画中画模式位置信息取值范围
    "sbs": [],           // 分屏模式位置信息取值范围
  },
}
```

请求方式

```
GET http://ip/usapi?method=set-video-mixer-config&input-device=1...
```

参数	说明
method	方法名称：set-video-mixer-config
input-device	格式 1: 与 SDI 输入一致 2: 与 HDMI 输入一致
is-hdmi-top	合成画面 Z 次序 0: HDMI 输入位于底层，SDI 输入位于顶层 1: HDMI 输入位于顶层，SDI 输入位于底层
type	位置类型 0: 画中画 1: 分屏
location	位置信息，通过 get-info 获取

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态，0：操作成功，返回其它值请参考 API 状态码

select-input-source 接口

设置音视频输入源。

通过 [get-info](#) 可以获取到录制相关参数的取值范围：

```
"input-source": {  
  "sources": [],           // 输入信号源取值范围  
}
```

请求方式

```
GET http://ip/usapi?method=select-input-source&input-source
```

参数	说明
method	方法名称: select-input-source
input-source	输入信号源 1: 只 SDI 输入 2: 只 HDMI 输入 3: SDI 或 HDMI 输入 4: SDI 和 HDMI 合成

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

get-signal-info 接口

获取视频和音频输入信号信息。

请求方式

```
GET http://ip/usapi?method=get-signal-info&input-device
```

参数	说明
method	方法名称: get-signal-info
input-device	输入信号源 1: SDI 输入 2: HDMI 输入

返回数据

```
{
  "result": 0
  "signal-info-types": [
    "video-info",
    "audio-info",
    "hdmi-info"
  ],
  "hdmi-info": {
    "mode": "hdmi",
    "vic": 0,
    "hdcpc": false,
    "it-content": false,
    "3d-format": false,
    "timing-h-total": 1650,
    "timing-h-active": 1280,
    "timing-h-frontporch": 110,
    "timing-h-syncwidth": 40,
    "timing-h-backporch": 220,
    "timing-f0v-syncwidth": 5,
    "timing-f0v-frontporch": 5,
    "timing-f0v-backproch": 20,
    "timing-f0v-active": 720,
    "timing-f0v-totalheight": 750
  },
  "audio-info": {
    "codec": "lpcm",
    "num-channels": 2,
    "sample-rate": 48000,
    "bit-count": 16
  },
  "video-info": {
    "codec": "uncompressed",
    "width": 1280,
    "height": 720,
    "scan": "progressive",
    "field-rate": 60.00,
    "color-depth": 8,
    "color-format": "bt.709",
    "aspect-ratio": "16:9",
    "sampling": "4:4:4",
    "quant-range": "limited",
    "sat-range": "limited",
    "frame-struct": "2d"
  }
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

set-video-color 接口

设置图像的亮度、对比度、色调、饱和度。

通过 [get-info](#) 可以获取到录制相关参数的取值范围：

```
"color-range": {
  "contrast": [],      // 图像对比度取值范围
  "brightness": [],   // 图像亮度取值范围
  "saturation": [],   // 图像饱和度取值范围
  "hue": []           // 图像色调取值范围
}
```

请求方式

```
GET http://ip/usapi?method=set-video-color&input-device=1&contrast=100&brightness=0&saturation=100&hue=0
```

参数	说明
method	方法名称: set-video-color
input-device	输入信号源 1: SDI 输入 2: HDMI 输入
contrast	设置对比度, 默认值: 100
brightness	设置亮度, 默认值: 0
saturation	设置饱和度, 默认值: 100
hue	设置色调, 默认值: 0

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-video-input-format 接口

设置视频输入信号的色彩空间和量化范围。

通过 [get-info](#) 可以获取相关参数的取值范围：

```
"video-format": {  
  "input-color-fmt": [],      // 色彩空间取值范围  
  "quant-range": [],        // 量化范围取值范围  
}
```

请求方式

```
GET http://ip/usapi?method=set-video-input-format&input-device=1&is-color-fmt=0&color-fmt=1&is-quant-range=0&quant-range=1
```

参数	说明
method	方法名称: set-video-input-format
input-device	输入信号源 1: SDI 输入 2: HDMI 输入
is-color-fmt	是否自定义设置色彩空间, 默认值: 0
color-fmt	设置色彩空间
is-quant-range	是否自定义设置量化范围, 默认值: 0
quant-range	设置量化范围

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-video-output-format 接口

设置视频输出的色彩空间和量化范围。

通过 [get-info](#) 可以获取相关参数的取值范围：

```
"video-format": {  
  "output-color-fmt": [],      // 色彩空间取值范围  
  "quant-range": [],         // 量化范围取值范围  
  "sat-range": [],          // 饱和范围取值范围  
}
```

请求方式

```
GET http://ip/usapi?method=set-video-output-format&input-device=1&is-color-fmt=0&color-fmt=1&is-quant-range=0&quant-range=1&is-sat-range=0&sat-range=1
```

参数	说明
method	方法名称: set-video-output-format
input-device	输入信号源 1: SDI 输入 2: HDMI 输入
is-color-fmt	是否自定义设置色彩空间, 默认值: 0
color-fmt	设置色彩空间, 默认值: 3
is-quant-range	是否自定义设置量化范围, 默认值: 0
quant-range	设置量化范围, 默认值: 2
is-sat-range	是否自定义设置饱和范围, 默认值: 0
sat-range	设置饱和范围, 默认值: 2

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-video 接口

配置主编码流参数和次编码流参数，相关参数如下：

- 视频分辨率
- 帧间隔
- 视频编码类型：主编码流可以选择 H264 或 HEVC，次编码流默认必须选择 H264。
- 编码质量
- 视频码率
- 关键帧间隔
- 量化范围
- 位率统计周期

通过 [get-info](#) 可以获取到各个参数的取值范围：

```
{
  "codec-cap": {
    "resolutions": [],           // 视频分辨率取值范围
    "durations": [],           // 帧间隔取值范围
    "video-codec": [],         // 视频编码类型取值范围
    "profile": [],             // H264 编码质量取值范围
    "hevc-profile": [],        // HEVC 编码质量取值范围
    "video-kbps": [],          // 视频编码码率取值范围
    "gop-sec": [],             // 关键帧间隔取值范围
    "video-range": [],         // 量化范围取值范围
    "stat-sec": []             // 位率统计周期取值范围
  }
}
```

请求方式

```
GET http://ip/usapi?method=set-video&stream=0&is-auto=0&cx=1280&cy=720&duration=333333&kbps=1024&gop=1&fourcc=0&profile=2&cbrstat=60&fullrange=1&codec=0
```

参数	说明
method	方法名称：set-video
stream	编码流类型 0：主编码流 1：次编码流
is-auto	设置方式，配置次编码流时只能为0 0：手动 1：自动
cx	分辨率宽度像素值
cy	分辨率高度像素值
duration	帧间隔
kbps	视频编码码率
gop	关键帧间隔
fourcc	保留参数。默认值：0
codec	编码类型
profile	编码质量
cbrstat	保留参数。
fullrange	量化范围
is-vbr	码率控制模式 0：CBR 1：VBR,最大最小量化步长必填
min-vbr-qp	最小量化步长，有效值:0 ~ 51
max-vbr-qp	最大量化步长，有效值:最小量化步长 ~ 51
is-time-code-sei	设置时间码 SEI 信息 0：不写入

is-time-code-sei	1: 写入系统时间 2: 写入输入源时间码
is-closed-caption-sei	是否设置隐藏式字幕 SEI 信息 0: 否 1: 是
ar-convert-mode	宽高比转换模式 0: 充满画面, 忽略宽高比 1: 裁剪部分画面, 保持宽高比 2: 填充黑边, 保持宽高比

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态。0: 操作成功, 返回其它值请参考 API 状态码

set-audio 接口

配置音频编码参数。

通过 `get-info` 可以获取到音频编码相关参数取值范围：

```
{
  "codec-cap": {
    "audio-streams": [],      // 音频码流取值范围
    "audio-channels": [],    // 音频编码通道数取值范围
    "audio-chmap": [],      // 音频编码通道映射取值范围
    "audio-kbps": [],        // 音频编码码率取值范围
  }
}
```

请求方式

```
GET http://ip/usapi?method=set-audio&stream=1&kbps=48...
```

参数	说明
method	方法名称: set-audio
stream	音频码流类型 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
channels	音频编码通道数
kbps	音频编码码率
ch0	音频编码 1 和 2 通道映射源通道序号
ch1	音频编码 3 和 4 通道映射源通道序号
ch2	音频编码 5 和 6 通道映射源通道序号
ch3	音频编码 7 和 8 通道映射源通道序号

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

enable-deinterlace 接口

设置视频去隔行。

请求方式

```
GET http://ip/usapi?method=enable-deinterlace&enable-deinterlace=1
```

参数	说明
method	方法名称: enable-deinterlace
enable-deinterlace	是否去隔行 0: 不去隔行 1: 去隔行

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

add-server 接口

新增直播服务器，最多可同时录入16条直播配置。

配置相同的直播仅可录入一个：

- 自定义类型 RTMP 直播，URL 和 StreamKey 都相同时为同一配置
- 其它类型 RTMP 直播，StreamKey 相同时为同一配置
- RTSP 直播，端口号相同时为同一配置
- SRT Caller 直播，IP 地址、端口号和流 ID 相同时为同一配置
- SRT Listener 直播，端口号相同时为同一配置
- HLS 直播，主码流或次码流流名称相同时为同一配置
- TS over UDP 与 TS over RTP 直播，IP 地址与端口号相同时为同一配置
- NDI HX 直播，视频源名称相同时为同一配置

请求方式

```
GET http://ip/usapi?method=add-server&type=xxx&url=xxx&key=xxx&...
```

参数	说明
method	方法名称：add-server
type	服务器类型 0: RTMP 1: Twitch 2: YouTube 3: Facebook 100: RTSP 120: SRT Caller 121: SRT Listener 130: NDI HX 131: HLS 132: TS over UDP 133: TS over RTP 140: TVU ISSP
name	服务器名称，字符要求： 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 _-+[]() 组成，并且不能以空格开头或结束
RTMP直播参数	
stream-index	编解码流 0: 主码流 1: 次码流
url	服务器地址
key	流密钥
is-auth	认证状态 0: 无需认证 1: 需要认证，用户名和密码必填
user	用户名
passwd	密码
token	Token
event-data	直播事件
net-mode	网络优先 0: 移动宽带优先 1: 有线网络优先 2: 无线网络优先
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
RTSP直播参数	
port	端口号，默认为554
max-connection	客户端最大连接数

is-auth	认证状态 0: 无需认证 1: 需要认证, 用户名和密码必填
user	用户名
passwd	密码
is-main	是否启用主码流 0: 不启用 1: 启用
main-stream-name	主码流名称, 字符要求: 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 . _ - + [] () 组成, 并且不能以空格开头或结束
is-sub	是否启用次码流 0: 不启用 1: 启用
sub-stream-name	次码流名称, 字符要求: 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 . _ - + [] () 组成, 并且不能以空格开头或结束
is-audio	是否启用音频 0: 不启用 1: 启用
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
SRT Caller直播参数	
stream-index	编码码流 0: 主码流 1: 次码流
url	服务器地址
port	端口号
latency	延迟时间,通过 get-info 可以获取相关参数的取值范围
bandwidth	带宽限制,通过 get-info 可以获取相关参数的取值范围
aes	加密方式 0: 不加密 16: AES-128加密 24: AES-192加密 32: AES-256加密
ase-word	加密字段
stream-id	选填
mtu	最大传输单元, 有效值:232 ~ 1500
conn-timeout	连接超时时间, 单位毫秒
retry-duration	等待重新连接时间, 单位毫秒
net-mode	网络优先 0: 移动宽带优先 1: 有线网络优先 2: 无线网络优先
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
SRT Listener直播参数	
stream-index	编码码流 0: 主码流 1: 次码流
port	端口号, 默认为8000
max-connection	客户端最大连接数
latency	延迟时间,通过 get-info 可以获取相关参数的取值范围
bandwidth	带宽限制,通过 get-info 可以获取相关参数的取值范围
aes	加密方式 0: 不加密 16: AES-128加密

	24: AES-192加密 32: AES-256加密
ase-word	加密字段
mtu	最大传输单元, 有效值:232 ~ 1500
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
NDI HX直播参数	
source-name	视频源名称, 默认为设备序列号
group-name	组名, 默认为public
enable-discovery	是否启用发现服务
discovery-server	发现服务器 IP 地址
transport-mode	传输模式, 通过 get-info 可以获取相关参数的取值范围; 选择组播传输模式, 组播相关设置必填
mcast-addr	组播地址
mcast-mask	组播掩码
mcast-ttl	生存时间,有效值:1 ~ 255
enable-fail-over	是否启用备用通道
fail-over-ndi-name	备用通道视频源名称
fail-over-ip-addr	备用通道 IP 地址
enable-web-control	是否显示 Web 控制
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
HLS直播参数	
stream-index	编解码流 0: 主码流 1: 次码流
stream-name	码流名称, 字符要求: 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 . _ - + []() 组成, 并且不能以空格开头或结束
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
TS over UDP / TS over RTP直播参数	
stream-index	编解码流 0: 主码流 1: 次码流
url	服务器地址
port	端口号
net-mode	网络优先 0: 移动宽带优先 1: 有线网络优先 2: 无线网络优先
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
TVU ISSP直播参数	
port	端口号, 默认为6539
stream-index	编解码流 0: 主码流 1: 次码流
audio	音频码流 0: 音频码流1 1: 音频码流2

2: 音频码流3
3: 音频码流4

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

enable-server 接口

新增服务器默认是禁用状态，只有通过该接口启用之后，才可以进行直播推流。

请求方式

```
GET http://ip/usapi?method=enable-server&id=1&is-use=1
```

参数	说明
method	方法名称: enable-server
id	服务器 ID, 通过 get-settings 获取
is-use	启用状态 0: 禁用 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

set-server 接口

修改直播服务器。

请求方式

```
GET http://ip/usapi?method=set-server&id=xxx&type=xxx&url=xxx&key=xxx&...
```

参数	说明
method	方法名称: set-server
id	服务器 ID, 通过 get-settings 获取
type	服务器类型 0: RTMP 1: Twitch 2: YouTube 3: Facebook 100: RTSP 120: SRT Caller 121: SRT Listener 130: NDI HX 131: HLS 132: TS over UDP 133: TS over RTP 140: TVU ISSP
name	服务器名称, 字符要求: 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 . _ - + []() 组成, 并且不能以空格开头或结束
RTMP直播参数	
stream-index	编码码流 0: 主码流 1: 次码流
url	服务器地址
key	流密钥
is-auth	认证状态 0: 无需认证 1: 需要认证, 用户名和密码必填
user	用户名
passwd	密码
token	Token
event-data	直播事件
net-mode	网络优先 0: 移动宽带优先 1: 有线网络优先 2: 无线网络优先
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
RTSP直播参数	
port	端口号, 默认为554
max-connection	客户端最大连接数
is-auth	认证状态 0: 无需认证 1: 需要认证, 用户名和密码必填
user	用户名
passwd	密码
is-main	是否启用主码流 0: 不启用 1: 启用
	主码流名称, 字符要求:

main-stream-name	1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 ._-+[]() 组成, 并且不能以空格开头或结束
is-sub	是否启用次码流 0: 不启用 1: 启用
sub-stream-name	次码流名称, 字符要求: 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 ._-+[]() 组成, 并且不能以空格开头或结束
is-audio	是否启用音频 0: 不启用 1: 启用
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
SRT Caller直播参数	
stream-index	编解码流 0: 主码流 1: 次码流
url	服务器地址
port	端口号
latency	延迟时间,通过 get-info 可以获取相关参数的取值范围
bandwidth	带宽限制,通过 get-info 可以获取相关参数的取值范围
aes	加密方式 0: 不加密 16: AES-128加密 24: AES-192加密 32: AES-256加密
ase-word	加密字段
stream-id	选填
mtu	最大传输单元, 有效值:232 ~ 1500
conn-timeout	连接超时时间, 单位毫秒
retry-duration	等待重新连接时间, 单位毫秒
net-mode	网络优先 0: 移动宽带优先 1: 有线网络优先 2: 无线网络优先
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
SRT Listener直播参数	
stream-index	编解码流 0: 主码流 1: 次码流
port	端口号, 默认为8000
max-connection	客户端最大连接数
latency	延迟时间,通过 get-info 可以获取相关参数的取值范围
bandwidth	带宽限制,通过 get-info 可以获取相关参数的取值范围
aes	加密方式 0: 不加密 16: AES-128加密 24: AES-192加密 32: AES-256加密
ase-word	加密字段
mtu	最大传输单元, 有效值:232 ~ 1500
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4

NDI HX直播参数	
source-name	视频源名称, 默认为设备序列号
group-name	组名, 默认为public
enable-discovery	是否启用发现服务
discovery-server	发现服务器 IP 地址
transport-mode	传输模式, 通过 get-info 可以获取相关参数的取值范围; 选择组播传输模式, 组播相关设置必填
mcast-addr	组播地址
mcast-mask	组播掩码
mcast-ttl	生存时间,有效值:1 ~ 255
enable-fail-over	是否启用备用通道
fail-over-ndi-name	备用通道视频源名称
fail-over-ip-addr	备用通道 IP 地址
enable-web-control	是否显示 Web 控制
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
HLS直播参数	
stream-index	编码码流 0: 主码流 1: 次码流
stream-name	码流名称, 字符要求: 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 . _ - ' [] () 组成, 并且不能以空格开头或结束
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
TS over UDP / TS over RTP直播参数	
stream-index	编码码流 0: 主码流 1: 次码流
url	服务器地址
port	端口号
net-mode	网络优先 0: 移动宽带优先 1: 有线网络优先 2: 无线网络优先
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4
TVU ISSP直播参数	
port	端口号, 默认为6539
stream-index	编码码流 0: 主码流 1: 次码流
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

del-server 接口

删除直播服务器。

请求方式

```
GET http://ip/usapi?method=del-server&id=1
```

参数	说明
method	方法名称: del-server
id	服务器 ID, 通过 get-settings 获取

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

start-test-server 接口

对服务器进行推流测试，同时间仅可对一路直播进行测试，测试过程中可以调用 [stop-test-server](#) 接口进行取消；当测试时间达到 20 秒时，自动停止直播测试。

请求方式

```
GET http://ip/usapi?method=start-test-server&type=xxx&url=xxx&key=xxx&...
```

参数	说明
method	方法名称: start-test-server
type	服务器类型 0: RTMP 1: Twitch 2: YouTube 3: Facebook 120: SRT Caller
name	服务器名称, 字符要求: 1、1 - 32 个字符 2、由 A-Z, a-z, 0-9, 空格 . _ - + []() 组成, 并且不能以空格开头或结束
RTMP直播参数	
stream-index	编码码流 0: 主码流 1: 次码流
url	服务器地址
key	流密钥
is-auth	认证状态 0: 无需认证 1: 需要认证, 用户名和密码必填
user	用户名
passwd	密码
token	Token
event-data	直播事件
net-mode	网络优先 0: 移动宽带优先 1: 有线网络优先 2: 无线网络优先
SRT Caller直播参数	
stream-index	编码码流 0: 主码流 1: 次码流
url	服务器地址
port	端口号
latency	延迟时间,通过 get-info 可以获取相关参数的取值范围
bandwidth	带宽限制,通过 get-info 可以获取相关参数的取值范围
aes	加密方式 0: 不加密 16: AES-128加密 24: AES-192加密 32: AES-256加密
ase-word	加密字段
stream-id	选填
net-mode	网络优先 0: 移动宽带优先 1: 有线网络优先 2: 无线网络优先

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态。 0: 开始测试 1: 操作重复 -9: 系统忙 返回其它值请参考 API 状态码

测试过程状态

通过 [get-status](#) 接口可以获取到 `living-test` 信息，如下：

```
"living-test": {
  "result": 27,
  "upload-bps": 0,
  "percent": 0,
  "net": 0,
  "client-id": ""
}
```

属性	说明
result	操作状态。 27: 初始状态 5: 操作取消 2: 测试中状态 0: 操作成功 返回其它值请参考 API 状态码
percent	测试进度：20 表示 20%
upload-bps	上传速度
net	网络类型
client-id	保留属性

进行测试后，需要调用 [clear-test-server](#) 接口将测试状态重置为初始状态（27）。

stop-test-server 接口

调用 [start-test-server](#) 接口后，服务器会进入推流测试过程，可以调用该接口取消操作。

请求方式

```
GET http://ip/usapi?method=stop-test-server
```

参数	说明
method	方法名称: stop-test-server

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功，返回其它值请参考 API 状态码

clear-test-server 接口

通过 [start-test-server](#) 进行推流测试，最后会处于测试成功或失败的状态，调用该接口可以重置为初始化状态 (27)。

请求方式

```
GET http://ip/usapi?method=clear-test-server
```

参数	说明
method	方法名称: clear-test-server

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

get-ndi-sources 接口

启用 NDIIHX 服务时，可以获取当前局域网内其他 NDI 发送端。

请求方式

```
GET http://ip/usapi?method=get-ndi-sources
```

参数	说明
method	方法名称: get-ndi-sources

返回数据

```
{
  "result": 0
  "ndi-sources": [
    {
      "ndi-name": "ULTRA ENCODE (A304220101001)",
      "ip-addr": "10.10.13.247:5962"
    }
  ]
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

set-ndi-find-config 接口

设置 NDIIHX 发现服务配置信息。

请求方式

```
GET http://ip/usapi?method=set-ndi-find-config&group-name=public&xxx
```

参数	说明
method	方法名称: set-ndi-find-config
group-name	组名, 默认为public
enable-discovery	是否启用发现服务
discovery-server	发现服务器 IP 地址

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

enable-ndi-hx3 接口

是否启用 NDI|HX3 直播服务。

请求方式

```
GET http://ip/usapi?method=enable-ndi-hx3&enable-ndi-hx3=1
```

参数	说明
method	方法名称: enable-ndi-hx3
enable-ndi-hx3	启用状态 0: 禁用 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

add-rec-channel 接口

新增录制通道，最多可同时录入3条录制配置，相同存储设备类型的录制通道只能添加一个。默认内置 U 盘、SD 卡和网络存储的录制通道。

请求方式

```
GET http://ip/usapi?method=add-rec-channel&type=xxx&stream-index=1...
```

参数	说明
method	方法名称: add-rec-channel
type	存储设备类型 0: U 盘 1: SD 卡 2: 网络存储
stream-index	编码码流 0: 主码流 1: 次码流
mode	录制模式 1: 普通录制 2: 循环录制
dir-name	录制目录, 默认值为 REC_Folder , 最多支持两级目录, 不同目录用 / 分割 (如: a/b) 。 字符要求: 由 A-Z, a-z, 0-9, 空格 _-+[]() 组成, 并且不能以空格或 / 开头或结束
file-prefix	录制文件名称前缀类型 0: 自定义 1: 以文件创建时间命名
prefix-name	录制文件名称前缀, 默认值为 VID , 字符要求: 由 A-Z, a-z, 0-9, 空格 _-+[]() 组成, 并且不能以空格开头或结束
file-suffix	录制视频格式 0: mp4 1: mov
time-unit	录制周期, 单位: 分钟。录制时, vfat 磁盘格式单个文件不得超过 4G, 即 time-unit * 录制编码率 <= 4G
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

enable-rec-channel 接口

启用录制通道，通过该接口启用之后，进行文件录制，最多可同时启用 2 路录制通道。

请求方式

```
GET http://ip/usapi?method=enable-rec-channel&id=1&is-use=1
```

参数	说明
method	方法名称: enable-rec-channel
id	录制通道 ID, 通过 get-rec-channels 获取
is-use	启用状态 0: 禁用 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

set-rec-channel 接口

修改录制通道。

请求方式

```
GET http://ip/usapi?method=set-rec-channel&id=xxx&type=xxx&stream-index=xxx&...
```

参数	说明
method	方法名称: set-rec-channel
id	录制通道 ID, 通过 get-rec-channels 获取
type	存储设备类型 0: U 盘 1: SD 卡 2: 网络存储
stream-index	编码码流 0: 主码流 1: 次码流
mode	录制模式 1: 普通录制 2: 循环录制
dir-name	录制目录, 默认值为 REC_Folder, 最多支持两级目录, 不同目录用 / 分割 (如: a/b)。 字符要求: 由 A-Z, a-z, 0-9, 空格, _-+[]() 组成, 并且不能以空格或 / 开头或结束
file-prefix	录制文件名称前缀类型 0: 自定义 1: 以文件创建时间命名
prefix-name	录制文件名称前缀, 默认值为 VID, 字符要求: 由 A-Z, a-z, 0-9, 空格, _-+[]() 组成, 并且不能以空格开头或结束
file-suffix	录制视频格式 0: mp4 1: mov
time-unit	录制周期, 单位: 分钟。录制时, vfat 磁盘格式单个文件不得超过 4G, 即 $time-unit * 录制编码率 \leq 4G$
audio	音频码流 0: 音频码流1 1: 音频码流2 2: 音频码流3 3: 音频码流4

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

del-rec-channel 接口

删除录制通道。

请求方式

```
GET http://ip/usapi?method=del-rec-channel&id=1
```

参数	说明
method	方法名称: del-rec-channel
id	录制通道 ID, 通过 get-rec-channels 获取

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

get-rec-channels 接口

获取所有录制通道配置信息。

请求方式

```
GET http://ip/usapi?method=get-rec-channels
```

参数	说明
method	方法名称: get-rec-channels

返回数据

```
{  
  "result": 0,  
  "rec-channels": [],  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

clear-rec-status 接口

启动录制后，会处于录制成功或失败的状态，调用该接口可以重置为初始化状态 (27)。

请求方式

```
GET http://ip/usapi?method=clear-rec-status&id=1
```

参数	说明
method	方法名称: clear-clear-rec-status
id	录制通道 ID, 通过 get-rec-channels 获取

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

add-nas 接口

新增网络存储服务器，最多可同时录入4条网络存储配置。

通过 [get-info](#) 可以获取到网络存储协议类型参数取值范围：

```
{
  "nas": {
    "proto": [],          // 网络存储协议类型参数取值范围
  }
}
```

请求方式

```
GET http://ip/usapi?method=add-nas&proto=xxx&addr=xxx&mount=xxx&...
```

参数	说明
method	方法名称：add-nas
proto	协议类型 0: NFS 1: CIFS
addr	服务器地址
mount	服务器挂载点
CIFS协议参数	
user	用户名
passwd	密码

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态，0：新增成功，返回其它值请参考 API 状态码

enable-nas 接口

启用网络存储服务器，将自动绑定到存储设备类型为网络存储的录制通道，最多可同时启用 1 个网络存储服务器。

请求方式

```
GET http://ip/usapi?method=enable-nas&id=1&is-use=1
```

参数	说明
method	方法名称: enable-nas
id	服务器 ID, 通过 get-settings 获取
is-use	启用状态 0: 禁用 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

set-nas 接口

修改网络存储服务器。

请求方式

```
GET http://ip/usapi?method=set-nas&id=xxx&proto=xxx&addr=xxx&mount=xxx&...
```

参数	说明
method	方法名称: set-nas
id	服务器 ID, 通过 get-settings 获取
proto	协议类型 0: NFS 1: CIFS
addr	服务器地址
mount	服务器挂载点
CIFS协议参数	
user	用户名
passwd	密码

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

del-nas 接口

删除网络存储服务器。

请求方式

```
GET http://ip/usapi?method=del-nas&id=1
```

参数	说明
method	方法名称: del-nas
id	服务器 ID, 通过 get-settings 获取

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

start-format-usb 接口

对 U 盘进行格式化，操作后数据无法恢复，并且格式化过程不可取消。

进行格式化时，必须满足以下条件：

- 未进行固件升级
- 未进行重置和重启
- 未启用录制功能
- 未对 U 盘进行性能测试
- U 盘有效

通过 `get-status` 接口可以获取到以下信息：

```
"cur-status": 65552 // 设备运行状态掩码
"disk-info" [
  {
    "disk-status": 1, // U 盘状态掩码
    "disk-type": 0, // 0: U 盘, 1: SD 卡
    "total-size": 61638148096,
    "used-size": 97058816,
    ...
  }
]
```

`cur-status` 和 [设备状态掩码](#) 进行与运算，可以得到以下状态表：

状态	条件
未进行固件升级	<code>cur-status & statusUpgrade != statusUpgrade</code>
未进行重置	<code>cur-status & statusReset != statusReset</code>
未进行重启	<code>cur-status & statusReboot != statusReboot</code>
未启用录制功能	<code>cur-status & statusRecord != statusRecord</code>
未对 U 盘进行性能测试	<code>cur-status & statusDiskTest != statusDiskTest</code>

`disk-status` 和 [U 盘状态码](#) 进行与运算，可以得到以下状态表：

状态	条件
U 盘有效	<code>disk-status & DiskValid = DiskValid</code>

请求方式

```
GET http://ip/usapi?method=start-format-usb
```

参数	说明
method	方法名称：start-format-usb

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态。 0: 开始格式化 1: 操作重复 -5: 磁盘错误 -9: 系统忙 返回其它值请参考 API 状态码

格式化过程状态

通过 `get-status` 接口可以获取到 `format-status` 信息，如下：

```
"format-status": {  
  "result": 27,  
  "percent": 20,  
  "client-id": ""  
}
```

属性	说明
result	操作状态。 27: 初始状态 2: 格式化中状态 0: 操作成功 返回其它值请参考 API 状态码
percent	格式化进度: 20 表示 20%
client-id	保留属性

U 盘格式化后，需要调用 `clear-format-usb` 接口将格式化状态重置为初始状态（27）。

start-format-sd 接口

对 SD 卡进行格式化，操作后数据无法恢复，并且格式化过程不可取消。

进行格式化时，必须满足以下条件：

- 未进行固件升级
- 未进行重置和重启
- 未启用录制功能
- SD 卡有效

通过 `get-status` 接口可以获取到以下信息：

```
"cur-status": 65552 // 设备运行状态掩码
"disk-info": [
  {
    "disk-status": 1, // SD 卡状态掩码
    "disk-type": 1, // 0: U 盘, 1: SD 卡
    "total-size": 61638148096,
    "used-size": 97058816,
    ...
  }
]
```

`cur-status` 和 [设备状态掩码](#) 进行与运算，可以得到以下状态表：

状态	条件
未进行固件升级	<code>cur-status & statusUpgrade != statusUpgrade</code>
未进行重置	<code>cur-status & statusReset != statusReset</code>
未进行重启	<code>cur-status & statusReboot != statusReboot</code>
未启用录制功能	<code>cur-status & statusRecord != statusRecord</code>

`disk-status` 和 [存储设备状态码](#) 进行与运算，可以得到以下状态表：

状态	条件
SD 卡有效	<code>disk-status & DiskValid = DiskValid</code>

请求方式

```
GET http://ip/usapi?method=start-format-sd
```

参数	说明
method	方法名称：start-format-sd

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态。 0: 开始格式化 1: 操作重复 -5: 磁盘错误 -9: 系统忙 返回其它值请参考 API 状态码

格式化过程状态

通过 `get-status` 接口可以获取到 `format-status` 信息，如下：

```
"format-status": {  
  "result": 27,  
  "percent": 20,  
  "client-id": ""  
}
```

属性	说明
result	操作状态。 27: 初始状态 2: 格式化中状态 0: 操作成功 返回其它值请参考 API 状态码
percent	格式化进度: 20 表示 20%
client-id	保留属性

SD 卡格式化后, 需要调用 [clear-format-sd](#) 接口将格式化状态重置为初始状态 (27)。

clear-format-usb 接口

U 盘进行格式化后，会处于格式化成功或失败的状态，调用该接口可以重置为初始化状态（27）。

请求方式

```
GET http://ip/usapi?method=clear-format-usb
```

参数	说明
method	方法名称：clear-format-usb

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0：操作成功，返回其它值请参考 API 状态码

clear-format-sd 接口

SD 卡进行格式化后，会处于格式化成功或失败的状态，调用该接口可以重置为初始化状态（27）。

请求方式

```
GET http://ip/usapi?method=clear-format-sd
```

参数	说明
method	方法名称：clear-format-sd

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0：操作成功，返回其它值请参考 API 状态码

start-test-usb 接口

对 U 盘的读写性能进行测试，判断是否满足当前录制码率的写入速度要求。测试过程中可以调用 `stop-test-usb` 接口进行取消。

进行性能测时，必须满足以下条件：

- 未进行固件升级
- 未进行重置和重启
- 未启用录制功能
- 未对 U 盘进行格式化
- 未对 U 盘进行性能测试
- U 盘有效
- U 盘空间大于 200 M
- U 盘可写入

通过 `get-status` 接口可以获取到以下信息：

```
"cur-status": 65552           // 设备运行状态掩码
"usb": {
  "disk-status": 1,           // U 盘状态掩码
  "total-size": 61638148096,
  "used-size": 97058816,
  ...
}
```

`cur-status` 和 [设备状态掩码](#) 进行与运算，可以得到以下状态表：

状态	条件
未进行固件升级	<code>cur-status & statusUpgrade != statusUpgrade</code>
未进行重置	<code>cur-status & statusReset != statusReset</code>
未进行重启	<code>cur-status & statusReboot != statusReboot</code>
未启用录制功能	<code>cur-status & statusRecord != statusRecord</code>
未对 U 盘进行格式化	<code>cur-status & statusFormatDisk != statusFormatDisk</code>
未对 U 盘进行性能测试	<code>cur-status & statusDiskTest != statusDiskTest</code>

`disk-status` 和 [U 盘状态码](#) 进行与运算，可以得到以下状态表：

状态	条件
U 盘有效	<code>disk-status & DiskValid = DiskValid</code>
U 盘空间大于 200 M	<code>disk-status & LowSpace != LowSpace</code>
U 盘可写入	<code>disk-status & NotWrite != NotWrite</code>

请求方式

```
GET http://ip/usapi?method=start-test-usb
```

参数	说明
method	方法名称：start-test-usb

返回数据

```
{
  "result": 0
}
```

属性	说明
	返回状态。 0: 开始格式化 1: 操作重复

result	-5: 磁盘错误 -9: 系统忙 -16: 磁盘空间不足 返回其它值请参考 API 状态码
--------	--

计算最低写入速度 requirSpeed

通过 [get-settings](#) 接口可以获取到录制配置信息，如下：

```

"rec-stream": 0,
"main-stream": {
  "kbps": 4096,
  ...
},
"sub-stream": {
  "kbps": 1024,
  ...
}
"audio": {
  "kbps": 128
  ...
}

```

rec-stream 表示录制码流类型：0 为 **main-stream**，1 为 **sub-stream**

通过以上信息可计算出当前录制时，requirSpeed 为：

```

requirSpeed = main-stream.kbps + audio.kbps
             = (4096 Kbps + 128 Kbps)
             = 4224 Kbps / 1024 * 8
             ≈ 0.52 MB/S

```

性能测试过程状态

通过 [get-status](#) 接口可以获取到 **disk-test** 信息，如下：

```

"disk-test": {
  "read-bps": 0,
  "write-bps": 14833071, // 14833071 bps / 1024 * 1024 ≈ 14.15 MB/S
  "percent": 3,
  "result": 2,
  "client-id": "web-session-32435088"
}

```

属性	说明
result	操作状态。 27: 初始状态 5: 操作取消 2: 测试中状态 0: 操作成功 返回其它值请参考 API 状态码
percent	格式化进度：20 表示 20%
read-bps	当前读取速度
write-bps	当前写入速度，如果大于 requirSpeed 则说明 U 盘性能是达标的
client-id	保留属性

进行性能测试后，需要调用 [clear-test-usb](#) 接口将测试状态重置为初始状态（27）。

stop-test-usb 接口

调用 `start-test-usb` 接口后，U 盘会进入性能测试过程，可以调用该接口取消操作。

请求方式

```
GET http://ip/usapi?method=stop-test-usb
```

参数	说明
method	方法名称: stop-test-usb

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功，返回其它值请参考 API 状态码

clear-test-usb 接口

U 盘通过 [start-test-usb](#) 进行性能测试，最后会处于测试成功或失败的状态，调用该接口可以重置为初始化状态（27）。

请求方式

```
GET http://ip/usapi?method=clear-test-usb
```

参数	说明
method	方法名称：clear-test-usb

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0：操作成功，返回其它值请参考 API 状态码

get-media-files 接口

获取 U 盘、SD 卡或网络存储中录制的文件。

获取到文件后，可以调用 [del-media-files](#) 接口进行删除。

请求方式

```
GET http://ip/usapi?method=get-media-files&disk-type=1&start=0&count=3
```

参数	说明
method	方法名称: get-media-files
disk-type	存储设备类型, 0: U 盘, 1: SD 卡, 2: 网络存储
start	起始索引, 从 0 开始
count	获取记录数量

返回数据

```
{
  "result": 0,
  "path": "/media/disk1/REC_Folder",
  "thumbnail": "thumbnail",
  "media-files": [
    {
      "name": "VID_9.mp4", // 视频文件名称
      "thumbnail-name": "VID_9.mp4_1569393321", // 缩略图名称, 默认后缀为 .jpg
      "status": 1, // 视频状态, 0 (正在录制), 1 (正常), 2 (错误), 3 (丢失)
      "create-time": "2019-09-25 06:35:21", // 创建时间
      "size-bytes": 4025670, // 视频大小, 单位: 字节
      "duration": 14480, // 视频时长, 单位: 秒
      "width": 1280, // 视频宽, 单位: 像素
      "height": 720, // 视频高, 单位: 像素
      "interval": 200000, // 帧率 (FPS) = 1000000 / interval
      "codec": 0 // 编码类型, 0: H264 1: HEVC
    },
    ...
  ]
}
```

属性	说明
result	返回状态。0: 操作成功, 返回其它值请参考 API 状态码
path	录制视频存储目录
thumbnail	视频缩率图存储子目录, 绝对路径为: path + thumbnail
media-files	文件列表

常用路径

名称	格式	示例
视频路径	path/name	/media/disk1/REC_Folder/VID_9.mp4
下载路径	http://ip:8080/download/path/name	http://ip:8080/download/media/disk1/REC_Folder/VID_9.mp4
图片路径	path/thumbnail/thumbnail-name	/media/disk1/REC_Folder/thumbnail/thumbnail-name.jpg

del-media-files 接口

批量删除 U 盘、SD 卡或网络存储中录制的文件。

请求方式

```
POST http://ip/usapi?method=del-media-files
```

参数	说明
method	方法名称: del-media-files

请求头

```
Content-Type: application/json;charset=UTF-8
```

Request Payload

```
{
  disk-type: 1,           // 存储设备类型, 0: U 盘, 1: SD 卡, 2: 网络存储
  media-files: ["VID_6_14.mp4"] // 要删除的文件名称数组
}
```

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态。0: 操作成功, 返回其它值请参考 API 状态码

add-image 接口

添加用于叠加的图片，仅支持分辨率小于 1920x1080，文件大小小于 512KB 的 JPEG 或 PNG 图片，最多支持添加 8 张。

请求方式

```
POST http://ip/usapi?method=add-image&name=xxx
```

参数	说明
method	方法名称: add-image
name	保留参数
enable-colorkey	保留参数
colorkey-mode	保留参数
key-color	保留参数
tolerance	保留参数

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

set-image 接口

修改图片，保留接口。

请求方式

```
GET http://ip/usapi?method=set-image&id=xxx
```

参数	说明
method	方法名称: set-image
id	录制通道 ID, 通过 get-settings 获取
name	保留参数
enable-colorkey	保留参数
colorkey-mode	保留参数
key-color	保留参数
tolerance	保留参数

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

del-image 接口

删除图片。

请求方式

```
GET http://ip/usapi?method=del-image&id=1
```

参数	说明
method	方法名称: del-image
id	图片 ID, 通过 get-images 获取

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

get-images 接口

获取添加的图片列表。

请求方式

```
GET http://ip/usapi?method=get-images
```

参数	说明
method	方法名称: get-images

返回数据

```
{  
  "result": 0,  
  "images": [],  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

add-surface 接口

新增一个叠加选项，主码流和次码流可以分别添加 8 个叠加选项，总计 16 个。

通过 [get-info](#) 可以获取到文字和图片叠加相关参数取值范围：

```
{
  "surface": {
    "image": {},           // 图片添加的限制条件
    "image-type": [],     // 图片类型的取值范围
    "type": [],           // 叠加类型的取值范围
    "position": [],       // 叠加位置的取值范围
    "font-family": [],    // 文字叠加字体的取值范围
    "font-style": [],     // 文字叠加样式的取值范围
    "font-weight": [],    // 文字叠加字重的取值范围
    "time-format": [],    // 时间叠加格式的取值范围
  }
}
```

请求方式

GET <http://ip/usapi?method=add-surface&type=xxx&stream-index=1...>

参数	说明
method	方法名称: add-surface
stream-index	编码码流 0: 主码流 1: 次码流
position	叠加位置
x-offset	叠加位置的左上角x坐标, 叠加位置为自定义时有效
y-offset	叠加位置的左上角y坐标, 叠加位置为自定义时有效
margin	叠加选项与外边距的宽度, 默认值为0
condition	保留参数
class	叠加选项类型 0: 文本 1: 系统时间 2: 图片
文本	
font-family	文字字体
font-size	字体大小
font-style	字体样式
font-weight	字体字重
font-color	字体颜色
line-width	边框宽度
line-color	边框颜色
width	文本宽度
height	文本高度
text	文本内容
系统时间	
font-family	文字字体
font-size	字体大小
font-style	字体样式
font-weight	字体字重
font-color	字体颜色
line-width	边框宽度

line-color	边框颜色
width	文本宽度
height	文本高度
text	时间格式
图片	
index	图片 ID, 通过 get-images 获取
alpha	透明度, 取值范围0-255
scale	缩放因子, 取值范围1-400%

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

set-enable-surface 接口

启动或停止主码流或次码流所有待叠加选项。

请求方式

```
GET http://ip/usapi?method=set-enable-surface&stream-index=1&enable=1
```

参数	说明
method	方法名称: set-enable-surface
stream-index	编码码流 0: 主码流 1: 次码流
enable	启用状态 0: 禁用 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

enable-surface 接口

启用叠加选项，进入待叠加状态，通过 [set-enable-surface](#) 启动所有待叠加选项。

请求方式

```
GET http://ip/usapi?method=enable-surface&id=1&is-use=1
```

参数	说明
method	方法名称: enable-surface
id	叠加选项 ID, 通过 get-surfaces 获取
is-use	启用状态 0: 禁用 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

set-surface 接口

修改叠加选项。

请求方式

```
GET http://ip/usapi?method=set-surface&id=xxx&stream-index=xxx&...
```

参数	说明
method	方法名称: set-surface
id	叠加选项 ID, 通过 get-surfaces 获取
stream-index	编码码流 0: 主码流 1: 次码流
position	叠加位置
x-offset	叠加位置的左上角x坐标, 叠加位置为自定义时有效
y-offset	叠加位置的左上角y坐标, 叠加位置为自定义时有效
margin	叠加选项与外边距的宽度, 默认值为0
condition	保留参数
class	叠加选项类型 0: 文本 1: 系统时间 2: 图片
文本	
font-family	文字字体
font-size	字体大小
font-style	字体样式
font-weight	字体字重
font-color	字体颜色
line-width	边框宽度
line-color	边框颜色
width	文本宽度
height	文本高度
text	文本内容
系统时间	
font-family	文字字体
font-size	字体大小
font-style	字体样式
font-weight	字体字重
font-color	字体颜色
line-width	边框宽度
line-color	边框颜色
width	文本宽度
height	文本高度
text	时间格式
图片	
index	图片 ID, 通过 get-images 获取
alpha	透明度, 取值范围0-255
scale	缩放因子, 取值范围1-400%

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

del-surface 接口

删除叠加选项。

请求方式

```
GET http://ip/usapi?method=del-surface&id=1
```

参数	说明
method	方法名称: del-surface
id	叠加选项 ID, 通过 get-surfaces 获取

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

get-surfaces 接口

获取所有叠加选项信息。

请求方式

```
GET http://ip/usapi?method=get-surfaces
```

参数	说明
method	方法名称: get-surfaces

返回数据

```
{
  "result": 0,
  "main-surface": 1,      // 主码流是否启用叠加
  "second-surface": 0,   // 次码流是否启用叠加
  "surfaces": [],
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

add-scheduler 接口

新增排程，最多为每一路直播或者录制通道同时录入 8 条排程配置。

请求方式

```
GET http://ip/usapi?method=add-scheduler&chn-type=xxx&chn-id=xxx...
```

参数	说明
method	方法名称: add-scheduler
chn-type	通道类型, 0: 直播通道 1: 录制通道
chn-id	直播或者录制通道 ID 直播通道 ID, 通过 get-settings 获取 录制通道 ID, 通过 get-rec-channels 获取
title	标题, 字符要求: 由 A-Z, a-z, 0-9, 空格 . _ - + [] () 组成, 并且不能以空格或 / 开头或结束
desc	描述, 字符要求: 由 A-Z, a-z, 0-9, 空格 . _ - + [] () 组成, 并且不能以空格或 / 开头或结束
is-full-day	是否全天运行 0: 不是全天 1: 全天
time-begin	开始时间, 时间格式: 2023-02-02 10:00:00
time-end	结束时间, 时间格式: 2023-02-02 11:00:00
is-repeat	是否重复事件 0: 不是 1: 是
repeat-event	重复时间类型 0: 不重复 1: 按日重复 2: 按周重复 3: 按月重复 4: 按年重复
按日重复	
repeat-interval	每几天重复
repeat-is-workday	每个工作日重复
按周重复	
repeat-is-interval	每几周重复
repeat-wdays	每周周几重复, 掩码, 可选多个 0x01: 星期一 0x02: 星期二 0x04: 星期三 0x08: 星期四 0x10: 星期五 0x20: 星期六 0x40: 星期日
按月重复	
repeat-interval	每几个月重复
repeat-mday	每几月的几日重复
repeat-wdy	每几月第几周重复
repeat-idx	每几月第几周的星期几重复 0: 星期一 1: 星期二 2: 星期三 3: 星期四 4: 星期五 5: 星期六 6: 星期日
按年重复	
repeat-month	每年的几月重复
repeat-mday	每年几月几日重复

repeat-wdy	每年几月第几周重复
repeat-idx	每年几月第几周的星期几重复 0: 星期一 1: 星期二 2: 星期三 3: 星期四 4: 星期五 5: 星期六 6: 星期日
end-event	结束时间类型 0: 无结束日期 1: 按重复次数结束 2: 按日期结束
按重复次数结束	
end-interval	重复几次结束
按日期结束	
end-dateline	重复事件结束日期

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

enable-scheduler 接口

启用排程。

请求方式

```
GET http://ip/usapi?method=enable-scheduler&id=1&is-use=1&...
```

参数	说明
method	方法名称: enable-scheduler
id	排程 ID, 通过 get-schedulers 获取
chn-type	通道类型, 0: 直播通道 1: 录制通道
chn-id	直播或者录制通道 ID 直播通道 ID, 通过 get-settings 获取 录制通道 ID, 通过 get-rec-channels 获取
is-use	启用状态 0: 禁用 1: 启用

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

set-scheduler 接口

修改排程。

请求方式

```
GET http://ip/usapi?method=set-scheduler&id=xxx&chn-type=xxx&chn-id=xxx&...
```

参数	说明
method	方法名称: set-scheduler
id	排程 ID, 通过 get-schedulers 获取
chn-type	通道类型, 0: 直播通道 1: 录制通道
chn-id	直播或者录制通道 ID 直播通道 ID, 通过 get-settings 获取 录制通道 ID, 通过 get-rec-channels 获取
title	标题, 字符要求: 由 A-Z, a-z, 0-9, 空格, _-+[]() 组成, 并且不能以空格或 / 开头或结束
desc	描述, 字符要求: 由 A-Z, a-z, 0-9, 空格, _-+[]() 组成, 并且不能以空格或 / 开头或结束
is-full-day	是否全天运行 0: 不是全天 1: 全天
time-begin	开始时间, 时间格式: 2023-02-02 10:00:00
time-end	结束时间, 时间格式: 2023-02-02 11:00:00
is-repeat	是否重复事件 0: 不是 1: 是
repeat-event	重复时间类型 0: 不重复 1: 按日重复 2: 按周重复 3: 按月重复 4: 按年重复
按日重复	
repeat-interval	每几天重复
repeat-is-workday	每个工作日重复
按周重复	
repeat-is-interval	每几周重复
repeat-wdays	每周周几重复, 掩码, 可选多个 0x01: 星期一 0x02: 星期二 0x04: 星期三 0x08: 星期四 0x10: 星期五 0x20: 星期六 0x40: 星期日
按月重复	
repeat-interval	每几个月重复
repeat-mday	每几月的几日重复
repeat-wdy	每几月第几周重复
repeat-idx	每几月第几周的星期几重复 0: 星期一 1: 星期二 2: 星期三 3: 星期四 4: 星期五 5: 星期六 6: 星期日
按年重复	
repeat-month	每年的几月重复

repeat-mday	每年几月几日重复
repeat-wdy	每年几月第几周重复
repeat-idx	每年几月第几周的星期几重复 0: 星期一 1: 星期二 2: 星期三 3: 星期四 4: 星期五 5: 星期六 6: 星期日
end-event	结束时间类型 0: 无结束日期 1: 按重复次数结束 2: 按日期结束
按重复次数结束	
end-interval	重复几次结束
按日期结束	
end-dateline	重复事件结束日期

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

del-scheduler 接口

删除排程。

请求方式

```
GET http://ip/usapi?method=del-scheduler&id=1&...
```

参数	说明
method	方法名称: del-scheduler
id	排程 ID, 通过 get-schedulers 获取
chn-type	通道类型, 0: 直播通道 1: 录制通道
chn-id	直播或者录制通道 ID 直播通道 ID, 通过 get-settings 获取 录制通道 ID, 通过 get-rec-channels 获取

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

get-schedulers 接口

获取排程配置信息。

请求方式

```
GET http://ip/usapi?method=get-schedulers&id=xxx&chn-type=xxx&chn-id=xxx
```

参数	说明
method	方法名称: get-schedulers
id	排程 ID, 通过 get-schedulers 获取, 值为-1则获取直播或录制通道的所有排程信息
chn-type	通道类型, 0: 直播通道 1: 录制通道
chn-id	直播或者录制通道 ID 直播通道 ID, 通过 get-settings 获取 录制通道 ID, 通过 get-rec-channels 获取

返回数据

```
{  
  "result": 0,  
  "schedulers": [],  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

reboot 接口

重启设备，重启后需要重新登录。

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

请求方式

```
GET http://ip/usapi?method=reboot
```

参数	说明
method	方法名称: reboot

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0 操作成功, 设备进入重启状态。返回其它值请参考 API 状态码

reset-all-settings 接口

将设备全部参数恢复至默认值。

重置时重新初始化网络，整个过程大概需要几分钟时间。可以使用 [ping 接口](#) 判断设备是否已经重置完成。

请求方式

```
GET http://ip/usapi?method=reset-all-settings
```

参数	说明
method	方法名称: reset-all-settings

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

login 接口

用户登录，登录成功后会在 Cookie 存放 Session ID (Cookie: sid=e0f6b33dd2b575eff40733b3778beaab)。

请求方式

```
GET http://ip/usapi?method=login&id=xxx&pass=xxx
```

参数	说明
method	方法名称: login
id	用户名
pass	密码, MD5 加密

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 登录成功, 36: 用户名或密码错误, 返回其它值请参考 API 状态码

logout 接口

退出登录, 返回到登录界面。

请求方式

```
GET http://ip/usapi?method=logout
```

参数	说明
method	方法名称: logout

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 退出登录成功, 返回其它值请参考 API 状态码

get-users 接口

获取系统用户列表信息，仅管理员有权限。

请求方式

```
GET http://ip/usapi?method=get-users
```

参数	说明
method	方法名称: get-users

返回数据

```
{
  "result": 0,
  "users": [
    {
      "id": "Admin",
      "type": 1
    },
    {
      "id": "Test",
      "type": 2
    }
  ]
}
```

属性	说明
result	返回状态, 0: 数据获取成功, 返回其它值请参考 API 状态码
users	用户列表数组, id: 用户名, type: 类型, 1 为管理员

add-user 接口

添加用户，仅管理员有权限。

请求方式

```
GET http://ip/usapi?method=add-user&id=xxx&pass=xxx
```

参数	说明
method	方法名称: add-user
id	用户名
pass	密码, MD5 加密

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 新增成功, 返回其它值请参考 API 状态码

del-user 接口

删除用户，仅管理员有权限。

请求方式

```
GET http://ip/usapi?method=del-user&id=xxx
```

参数	说明
method	方法名称: del-user
id	用户登录名

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 删除成功, 返回其它值请参考 API 状态码

ch-passwd 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

```
GET http://ip/usapi?method=ch-passwd&pass=xxx&new-pass=xxx
```

参数	说明
method	方法名称: ch-passwd
pass	原密码, MD5 加密
new-pass	新密码, MD5 加密

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 修改成功, 返回其它值请参考 API 状态码

set-passwd 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

```
GET http://ip/usapi?method=set-passwd&id=xxx&pass=xxx
```

参数	说明
method	方法名称: set-passwd
id	用户登录名
pass	新密码, MD5 加密

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 重置成功, 返回其它值请参考 API 状态码

set-net 接口

设置以太网和 Wi-Fi 配置信息。

请求方式

```
GET http://ip/usapi?method=set-net&type=0&is-dhcp=1&ip=10.10.107.212&mask=255.255.0.0&router=10.10.0.1&dns=10.0.0.3
```

参数	说明
method	方法名称: set-net
type	网口类型 0: 无线网卡 1: 有线网卡
is-dhcp	是否动态获取 IP 地址 0: 手动设置 1: 动态获取
ip	IP 地址
mask	子网掩码
router	网关
dns	DNS 服务器

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

scan-wifi-results 接口

获取可连接的无线网络信息。

请求方式

```
GET http://ip/usapi?method=scan-wifi-results
```

参数	说明
method	方法名称: scan-wifi-results

返回数据

```
{
  "result": 0
  "user-items": [
  ],
  "scan-items": [
  ]
}
```

已连接无线网络

```
"user-items": [
  {
    "name": "",
    "freq": 5785,
    "level": -35,
    "is-auto": 0,
    "is-use": 0
  }
]
```

未连接无线网络

```
"scan-items": [
  {
    "name": "",
    "freq": 5785,
    "level": -35,
    "is-auto": 0,
    "is-use": 0
  }
]
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

connect-wifi-first 接口

设备首次连接此无线网络。

请求方式

```
GET http://ip/usapi?method=connect-wifi-first&name=magewell&passwd=1111111&secu=2&is-auto=1
```

参数	说明
method	方法名称: connect-wifi-first
name	无线网络的名称
passwd	无线网络的密码
secu	无线网络的加密方式, 0: unencrypted 1: wep 2: WPA/PSK 3: WPA2/PSK
is-auto	是否允许自动连接此无线网络

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

connect-wifi 接口

设备连接已保存的无线网络。

请求方式

```
GET http://ip/usapi?method=connect-wifi&name=magewell&is-auto=1
```

参数	说明
method	方法名称: connect-wifi
name	无线网络的名称
is-auto	是否允许自动连接此无线网络

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

cancel-connect-wifi 接口

取消正在连接的无线网络。

请求方式

```
GET http://ip/usapi?method=cancel-connect-wifi&name=magewell
```

参数	说明
method	方法名称: cancel-connect-wifi
name	无线网络的名称

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

disconnect-wifi 接口

断开无线网络连接。

请求方式

```
GET http://ip/usapi?method=disconnect-wifi
```

参数	说明
method	方法名称: disconnect-wifi

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

forget-wifi 接口

忘记已保存的无线网络，设备删除此无线网络的相关信息。

请求方式

```
GET http://ip/usapi?method=forget-wifi&name=magewell
```

参数	说明
method	方法名称: forget-wifi
name	无线网络的名称

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-connect-wifi-auto 接口

设置是否允许设备自动连接此无线网络。

请求方式

```
GET http://ip/usapi?method=set-connect-wifi-auto&name=magewell&is-auto=1
```

参数	说明
method	方法名称: set-connect-wifi-auto
name	无线网络的名称
is-auto	是否允许自动连接此无线网络

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

clear-connect-wifi 接口

设备通过 [connect-wifi-first](#) 配置无线网络，如果配置失败，调用该接口可以重置为初始化状态（27）。

请求方式

```
GET http://ip/usapi?method=clear-connect-wifi
```

参数	说明
method	方法名称：clear-connect-wifi

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0：操作成功，返回其它值请参考 API 状态码

close-softap 接口

开启 AP 模式。

请求方式

```
GET http://ip/usapi?method=open-softap
```

参数	说明
method	方法名称: open-softap

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

close-softap 接口

关闭 AP 模式。

请求方式

```
GET http://ip/usapi?method=close-softap
```

参数	说明
method	方法名称: close-softap

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

set-rndis-config 接口

设置 USB 网络配置信息。

请求方式

```
GET http://ip/usapi?method=set-rndis-config&addr=192.168.66.1
```

参数	说明
method	方法名称: set-rndis-config
addr	IP 地址

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

update 接口

固件更新有两种方式：

- 手动更新：用户可以先通过 [upload-update-file](#) 接口上传指定版本的固件，然后再进行更新
- 在线升级：当有新版本时，用户可以在线升级到最新版本，[online-update-check](#) 接口检测线上是否有新版本固件。

进行固件更新时，必须满足以下条件：

- 未进行检测新版本固件

通过 [get-status](#) 接口可以获取到以下信息：

```
"cur-status": 65552 // 设备运行状态掩码
```

`cur-status` 和 [设备状态掩码](#) 进行与运算，可以得到以下状态表：

状态	条件
未进行检测新版本固件	<code>cur-status & statusCheckUpgrade != statusCheckUpgrade</code>

请求方式

```
GET http://ip/usapi?method=update&mode=xxx
```

参数	说明
method	方法名称：update
mode	更新方式 upload：手动更新 online：在线升级

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态，0：开始更新，返回其它值请参考 API 状态码

升级过程状态

通过 [get-status](#) 接口可以获取到 `upgrade-status` 信息，如下：

```
"upgrade-status": {  
  "result": 27,  
  "step": 0,  
  "percent": 0,  
  "mode": "none",  
  "client-id": ""  
}
```

属性	说明
result	操作状态 27：初始状态 2：更新中状态 0：操作成功 返回其它值请参考 API 状态码
step	更新步骤
percent	格式化进度：20 表示 20%
mode	更新方式

client-id	保留属性
-----------	------

如果更新失败，需要调用 [clear-upgrade](#) 接口将更新状态重置为初始状态 (27)。

upload-update-file 接口

上传固件，上传文件格式为.mwf。

请求方式

```
POST http://ip/usapi?method=upload-update-file
```

参数	说明
method	方法名称: upload-update-file

返回数据

```
{
  "status": 0,
  "up-to-date": true,
  "version": "1.1.72",
  "size": 11890776
}
```

属性	说明
status	返回状态, 0: 上传成功, 返回其它值请参考 API 状态码
up-to-date	是否为最新版本, true/false
version	上传固件版本号
size	上传固件大小 (B)

固件上传完成后, 可以调用 [update](#) 接口进行更新。

cancel-download 接口

固件通过 [update](#) 进行在线升级时，会先下载新固件，下载过程可以调用该接口进行取消。

请求方式

```
GET http://ip/usapi?method=cancel-download
```

参数	说明
method	方法名称：cancel-download

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0：操作成功，返回其它值请参考 API 状态码

online-update-check 接口

在线检测是否有新版本的固件。

进行检测时，必须满足以下条件：

- 未进行固件升级

通过 `get-status` 接口可以获取到以下信息：

```
"cur-status": 65552 // 设备运行状态掩码
```

`cur-status` 和 [设备状态掩码](#) 进行与运算，可以得到以下状态表：

状态	条件
未进行固件升级	<code>cur-status & statusUpgrade != statusUpgrade</code>

请求方式

```
GET http://ip/usapi?method=online-update-check
```

参数	说明
method	方法名称：online-update-check

返回数据

```
{
  "result": 0
}
```

属性	说明
result	返回状态，0：操作成功，返回其它值请参考 API 状态码

检测过程状态

通过 `get-status` 接口可以获取到 `check-upgrade` 和 `upgrade` 信息，如下：

```
"check-upgrade": {
  "result": 0,
  "client-id": ""
},
"upgrade": {
  "ver": "1.2.123",
  "date": "2012-1-1 00:00:00",
  "size-byte": 12004784,
  "info": [
    {
      "version": "1.2.123",
      "changelog": "## Develop version 1.2. ## Develop version2."
    }
  ]
}
```

属性	说明
result	操作状态。 27：初始状态 2：检测中状态 0：操作成功 返回其它值请参考 API 状态码
ver	新固件版本号
date	新固件发布时间

size-byte	新固件文件大小 (B)
info	更新日志
client-id	保留属性

检测操作失败后，需要调用 [clear-check-update](#) 接口将检测状态重置为初始状态 (27)。

clear-upgrade 接口

固件通过 [update](#) 进行手动更新或在线升级后，如果更新失败，调用该接口可以重置为初始化状态（27）。

请求方式

```
GET http://ip/usapi?method=clear-upgrade
```

参数	说明
method	方法名称：clear-upgrade

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0：操作成功，返回其它值请参考 API 状态码

clear-check-update 接口

通过 [online-update-check](#) 进行检测新版本固件，最后会处于检测成功或失败的状态，调用该接口可以重置为初始化状态（27）。

请求方式

```
GET http://ip/usapi?method=clear-check-update
```

参数	说明
method	方法名称: clear-check-update

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功，返回其它值请参考 API 状态码

set-enable-check-update 接口

是否允许设备进行固件版本检测。

请求方式

```
GET http://ip/usapi?method=set-enable-check-update&is-check-update=1
```

参数	说明
method	方法名称: set-enable-check-update
is-check-update	是否允许 0: 不允许 1: 允许

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

import-edid 接口

为输入端口导入 EDID 文件。

请求方式

```
POST http://ip/usapi?method=import-edid
```

参数	说明
method	方法名称: import-edid

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 API 状态码

export-edid 接口

将输入端口 EDID 保存到本地。

请求方式

```
GET http://ip/usapi?method=export-edid&file-name=...
```

参数	说明
method	方法名称: export-edid
file-name	保存的文件路径

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

set-default-edid 接口

将输入端口 EDID 恢复到默认值。

请求方式

```
GET http://ip/usapi?method=set-default-edid
```

参数	说明
method	方法名称: set-default-edid

返回数据

```
{
  "result": 0
  "data": "AP////////wA09wEAAQAAAAEaAQOAAAB4Au6Vo1RMmSYPUFT//4AxQEVAYUBxQIGA0QDhwAEAC0gAMPJwWoCwWIoAUB10AAAEAjqAG
HE4LUBYLEUAUB10AAAEAAAA/QAP1g+HPAAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdX19gZWZiY2QHFgMSMgl/B
xUHUD0GwFcGAF9/AWd/AINPAADiAA9uAwwAEAC4eCEQgAECaWn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUQAeMEaPMwBQHxQAA
B4AAAAAAAAAAAAzw=="
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

get-edid-config 接口

获取输入端口 EDID 配置信息。

请求方式

```
GET http://ip/usapi?method=get-edid-config
```

参数	说明
method	方法名称: get-edid-config

返回数据

```
{
  "result": 0,
  "smart-edid": 1,
  "keep-last": 0,
  "add-audio": 1,
  "limit-pixel-clock": 1,
  "data": "AP////////wA09wEAAQAAAAEaAQOAAAB4Au6Vo1RMmSYPUFT//4AxQEVAyUBxQIGA0QDhwAEAC0gAMPJwWoCwWIoAUB10AAAEAjqAG
HE4LUBYLEUAUB10AAAEAAAA/QAPlg+HPAAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdX19gZWZiY2QHFgMSMgl/B
xUHUD0GwFcGAF9/Awd/AINPAADiAA9uAwwAEAC4eCEQgAECaWrn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUQAeMEaPMwBQHxQAA
B4AAAAAAAAAAAAzw=="
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

set-edid-config 接口

修改输入端口 EDID 配置信息。

请求方式

```
GET http://ip/usapi?method=set-edid-config&...
```

参数	说明
method	方法名称: set-edid-config
smart-edid	是否启用 SmartEDID 0: 否 1: 是
keep-last	是否保持最后一次的环出 EDID 0: 否 1: 是
add-audio	是否加入音频 0: 否 1: 是
limit-pixel-clock	是否使用限制像素时钟 0: 否 1: 是

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

get-loop-through-edid 接口

获取环出端口 EDID 信息。

请求方式

```
GET http://ip/usapi?method=get-loop-through-edid
```

参数	说明
method	方法名称: get-loop-through-edid

返回数据

```
{
  "result": 0
  "data": "AP////////wA09wEAAQAAAAEaAQOAAAB4Au6Vo1RMmSYPUFT//4AxQEVAYUBxQIGA0QDhwAEAC0gAMPJwWoCwWIoAUB10AAAEAjqAG
HE4LUBYLEUAUB10AAAEAAAA/QAP1g+HPAAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdX19gZWZiY2QHFgMSMgl/B
xUHUD0GwFcGAF9/AWd/AINPAADiAA9uAwwAEAC4eCEQgAECARn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUQAeMEaPMwBQHxQAA
B4AAAAAAAAAAAAzw=="
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

export-loop-through-edid 接口

将环出端口 EDID 保存到本地。

请求方式

```
GET http://ip/usapi?method=export-loop-through-edid&file-name=...
```

参数	说明
method	方法名称: export-loop-through-edid
file-name	保存的文件路径

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态。 0: 操作成功, 返回其它值请参考 API 状态码

cloud-reg-ex 接口

注册到云平台，默认可以注册到 2 个云平台服务器。

请求方式

```
GET http://ip:8070/cloud-api?method=cloud-reg-ex&id=1&cloud-enable-https=0&...
```

参数	说明
method	方法名称: cloud-reg-ex
id	云平台序号, 有效值: 0 或 1
cloud-code	邀请码, 4 位数字组成的字符串
cloud-ip-addr	服务器地址
cloud-http-port	http 服务器端口
cloud-enable-https	是否启用 https 服务 0: 不启用 1: 启用
cloud-https-port	https 服务器端口

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 云平台 API 状态码

cloud-unreg-ex 接口

取消注册到云平台。

请求方式

```
GET http://ip:8070/cloud-api?method=cloud-unreg-ex&id=1
```

参数	说明
method	方法名称: cloud-reg-ex
id	云平台序号, 有效值: 0 或 1

返回数据

```
{  
  "result": 0  
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 云平台 API 状态码

cloud-status 接口

获取云平台服务器状态。

请求方式

GET http://ip:8070/cloud-api?method=cloud-status&version=1

参数	说明
method	方法名称: cloud-status
version	云平台版本, 有效值: 1

返回数据

```
{
  "device_id": "B313221201001", // 设备序列号
  "number": 2, // 支持云平台数量
  "version": 1,
  "result": 0,
  "status": [
    {
      "cloud-code": "",
      "cloud-date": 0,
      "cloud-enable-https": 0,
      "cloud-http-port": 80,
      "cloud-https-port": 443,
      "cloud-ip-addr": "10.0.1.32",
      "cloud-reg-status": 101,
      "cloud-status": 35,
      "id": 0,
      "is-cloud-set": 1
    },
    {
      "cloud-code": "",
      "cloud-date": 0,
      "cloud-enable-https": 0,
      "cloud-http-port": 80,
      "cloud-https-port": 443,
      "cloud-ip-addr": "10.10.8.233",
      "cloud-reg-status": 103,
      "cloud-status": 35,
      "id": 1,
      "is-cloud-set": 1
    }
  ]
}
```

属性	说明
result	返回状态, 0: 操作成功, 返回其它值请参考 云平台 API 状态码