

Ultra Encode

前言

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

通用接口

get-info	2.1
get-status	2.2
get-settings	2.3
ping	2.4
get-signal-info	2.5
get-report	2.6
export-report	2.7

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-video-color	3.6
set-video-input-format	3.7
set-video-output-format	3.8
set-sync-offset	3.9

直播服务器

add-server	4.1
enable-server	4.2
set-server	4.3
del-server	4.4
start-test-server	4.5
stop-test-server	4.6
clear-test-server	4.7
get-ndi-sources	4.8

编码参数

set-video	5.1
set-audio	5.2

设备重置

reboot	6.1
reset-all-settings	6.2

用户登录

login	7.1
logout	7.2

用户管理

get-users	8.1
add-user	8.2
del-user	8.3
ch-password	8.4
set-password	8.5

网络设置

set-eth	9.1
scan-wifi-results	9.2
connect-wifi-first	9.3
connect-wifi	9.4
cancel-connect-wifi	9.5
disconnect-wifi	9.6
forget-wifi	9.7
set-connect-wifi-auto	9.8
clear-connect-wifi	9.9
open-softap	9.10
close-softap	9.11

固件更新

update	10.1
upload-update-file	10.2
cancel-download	10.3
online-update-check	10.4
clear-upgrade	10.5
clear-check-update	10.6
set-enable-check-update	10.7

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 HDMI
- Ultra Encode SDI

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=xxxxxxxx

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 A304201201001",
  ...
}
```

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
}
```

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,    // 保留
  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,  // 保留
  statusFormatSD   = 0x200000,  // 保留
  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 A304201201001",
  "serialnumber": "A304201201001",
  "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_hdmi_rev_a_1_3_328.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": "70:b3:d5:75:d0:4c",
    "wifi": "70:b3:d5:75:d0:4d",
    "blue": "70:b3:d5:75:d0:4e"
  },
  "snapshot": "/tmp/sbox-snapshot/sbox-quarter.jpg",
  "product": {
    "sn": "A304201201001",
    "product-id": 772,
    "hardware-ver": "A",
    "firmware-id": 1,
    "firmware-ver-s": "1.3.328",
    "factory-firmware-ver-s": "1.3.328",
    "product-name": "Ultra Encode",
    "module-name": "Ultra Encode HDMI",
    "manu-name": "MAGEWELL",
    "features": 1,
    "max-lock-count": 2
  },
  "audio-range": {
    "hdmi": {
      "max": 6.00,
      "min": -100.00,
      "def": 0.00
    },
    "mic": {
      "max": 55.25,
      "min": -12.00,
      "def": 0.00
    },
    "phone": {
```

```
        "max": 6.00,  
        "min": -57.00,  
        "def": 0.00  
    }  
},  
"codec-cap": {}  
}
```

***** 3. upload firmware *****

upload firmware response data:

```
{  
    "result": 0,  
    "up-to-date": true,  
    "version": "1.2.123",  
    "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
  "product": {}, // 产品相关信息
  "audio-range": { // 音频相关参数取值范围
    "hdm": {},
    "mic": {},
    "phone": {}
  },
  "codec-cap": { // 编码相关参数取值范围
    "resolutions": [],
    "durations": [],
    "profile": [],
    "hevc-profile": [],
    "video-kbps": [],
    "audio-kbps": [],
    "gop-sec": [],
    "video-range": [],
    "stat-sec": [],
    "video-codec": []
  },
  "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": {
  "max-bandwidth": 16384,
  "srt": {
    "latency": [],
    "bandwidth": [],
    "aes": []
  }
}
}
```

返回示例

```
"result": 0,
  "mac-addr": {
    "eth": "d0:c8:57:80:2d:7c",
    "wifi": "d0:c8:57:80:2d:7d",
```

```
"blue": "d0:c8:57:80:2d:7e"
},
"snapshot": "/tmp/sbox-snapshot/sbox-quarter.jpg",
"product": {
  "sn": "A304200731004",
  "product-id": 772,
  "hardware-ver": "A",
  "firmware-id": 1,
  "firmware-ver-s": "1.3.325",
  "factory-firmware-ver-s": "1.3.237",
  "product-name": "Ultra Encode",
  "module-name": "Ultra Encode HDMI",
  "manu-name": "MAGEWELL",
  "features": 1,
  "max-lock-count": 2
},
"audio-range": {
  "spi": {
    "max": 6.00,
    "min": -100.00,
    "def": 0.00
  },
  "linein": {
    "max": 6.00,
    "min": -100.00,
    "def": 0.00
  },
  "lineout": {
    "max": 6.00,
    "min": -57.00,
    "def": 0.00
  }
},
"codec-cap": {
  "resolutions": [
    {
      "w": 480,
      "h": 270
    },
    {
      "w": 480,
      "h": 360
    },
    {
      "w": 640,
      "h": 360
    },
    {
      "w": 640,
      "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": 1200
  },
  {
    "w": 1920,
    "h": 1080
  }
],
"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
}
],
"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
  }
]
],
"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,
    "srt": {
      "latency": [
        {
          "name": "30 ms",
          "value": 30
        },
        {
          "name": "60 ms",
          "value": 60
        },
        {
          "name": "120 ms",
          "value": 120
        },
        {
          "name": "500 ms",
          "value": 500
        },
        {
          "name": "1000 ms",
          "value": 1000
        },
        {
          "name": "5000 ms",
          "value": 5000
        }
      ]
    }
  }
}

```

```
    },
    {
      "name": "8000 ms",
      "value": 8000
    }
  ],
  "bandwidth": [
    {
      "name": "10%",
      "value": 10
    },
    {
      "name": "25%",
      "value": 25
    },
    {
      "name": "50%",
      "value": 50
    },
    {
      "name": "75%",
      "value": 75
    }
  ],
  "aes": [
    {
      "name": "Not Used",
      "value": 0
    },
    {
      "name": "AES-128",
      "value": 16
    },
    {
      "name": "AES-192",
      "value": 24
    },
    {
      "name": "AES-256",
      "value": 32
    }
  ]
}
}
```

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
  "sysstat": {}, // 设备运行状态
  "live-status": { // 直播状态
    "live": []
  },
  "upgrade-status": {}, // 固件升级过程状态
  "living-test": {}, // 直播推流测试过程状态
  "check-upgrade": {}, // 直播推流测试过程状态
  "conn-wifi": {}, // 无线网络连接过程状态
  "input-signal": {}, // 输入信号信息
  "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
  "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  
}
```

直播状态

```
"live-status": {  
  [  
    {  
      "result": 0,  
      "run-ms": 0,  
      "cur-bps": 0,  
      "avg-bps": 0,  
      "net": 0,  
      "result2": 0,  
      "cur-bps2": 0,  
      "net2": 0,  
      "client-id": ""  
    }  
  ]  
}
```

直播推流测试过程状态

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

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)
  "softap": {},              // AP 配置信息
  "video-color": {},         // 视频相关信息
  "volume": {},              // 音频相关信息
  "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
  "eth": {},                 // 以太网信息
  "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
}
```

返回示例:

```
{
  "result": 0,
  "name": "Ultra Encode A304201201001",
  "passwd": 1,
  "is-check-update": 1,
  "audio-sync-offset": 0,
  "softap": {
    "is-softap": 1,
    "is-visible": 1,
    "softap-ssid": "A304201201001",
    "softap-passwd": "01201001"
  },
  "video-color": {
    "contrast": 100,
    "brightness": 0,
    "saturation": 100,
    "hue": 0
  },
  "volume": {
    "is-mic": 1,
    "mic-gain": 0,
    "is-spi": 1,
    "spi-gain": 0,
    "is-phone": 1,
    "phone-gain": 0
  },
  "main-stream": {
    "is-auto": 0,
    "codec": 0,
    "cx": 1920,
    "cy": 1080,
  }
}
```

```

    "duration": 166667,
    "kbps": 4096,
    "gop": 1,
    "fourcc": 0,
    "profile": 2,
    "cbrstat": 60,
    "fullrange": 0
  },
  "sub-stream": {
    "cx": 1024,
    "cy": 768,
    "duration": 333333,
    "kbps": 1024,
    "gop": 1,
    "fourcc": 0,
    "profile": 2,
    "cbrstat": 60,
    "fullrange": 1
  },
  "audio": {
    "sample-rate": 48000,
    "channels": 2,
    "kbps": 128
  },
  "eth": {
    "is-dhcp": 1,
    "ip": "",
    "mask": "",
    "router": "",
    "dns": ""
  },
  "stream-server": [
    {
      "id": 0,
      "type": 0,
      "url": "192.168.1.123:345/live",
      "key": "aa",
      "is-auth": 0,
      "user": "",
      "passwd": "",
      "is-use": 0,
      "token": "",
      "net-mode": 1,
      "name": "192.168.1.123"
    }
  ],
  "video-input-format": {
    "is-color-fmt": 0,
    "color-fmt": 3,
    "is-quant-range": 0,
    "quant-range": 2
  },
  "video-output-format": {
    "is-color-fmt": 0,
    "color-fmt": 3,
    "is-quant-range": 0,
    "quant-range": 2,
    "is-sat-range": 0,
    "sat-range": 2
  }
}

```


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,  
  "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-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
```

参数	说明
method	方法名称: get-signal-info

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

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

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

新增于: V1.3

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

新增于: V1.3

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

新增于: V1.3

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": {  
  "hdmi": [],      // 输入信号音量增益取值范围  
  "mic": [],       // 麦克风音量增益取值范围  
  "phone": [],    // 耳机音量增益取值范围  
}
```

请求方式

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

参数	说明
method	方法名称: set-volume
is-mic	设置麦克风音量增益启用状态, 0: 静音 1: 启用
mic-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-phone	设置耳机音量增益启用状态, 0: 静音 1: 启用
phone-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-video-color 接口

新增于: V1.3

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&contrast=100&brightness=0&saturation=100&hue=0
```

参数	说明
method	方法名称: set-video-color
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

新增于: V1.4

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&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
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

新增于: V1.4

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&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
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-sync-offset 接口

新增于: V1.3

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-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 地址与端口号相同时为同一配置
- 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

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: 自定义类型 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
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
RTSP直播参数	
port	端口号，默认为554
max-connection	客户端最大连接数
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
	是否启用次码流

is-sub	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
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
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 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
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
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-contro	是否显示 Web 控制
HLS直播参数	
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
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

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。

同时只允许启用一种直播协议，SRT Call 与 SRT Listener 为同一直播协议，TS over UDP 与 TS over RTP 为同一直播协议。默认情况下最大支持启用 6 路直播，但有以下情况时，只允许启用一路：

- 当直播是 RTSP 直播时
- 当直播是 NDI HX 直播时
- 当直播是 HLS 直播时

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: 自定义类型 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
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
RTSP直播参数	
port	端口号, 默认为554
max-connection	客户端最大连接数
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

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
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
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
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-contro	是否显示 Web 控制
HLS直播参数	
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
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

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: 自定义类型 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": "PRO CONVERT (#14 (B403190104002))",
      "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-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	位率统计周期
fullrange	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-audio 接口

配置音频编码参数。

通过 [get-info](#) 可以获取到音频编码码率取值范围：

```
{
  "codec-cap": {
    "audio-kbps": [],          // 音频编码码率取值范围
  }
}
```

请求方式

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

参数	说明
method	方法名称：set-audio
kbps	音频编码码率

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

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?/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-password 接口

用户修改自己的登录密码，修改时必须输入原密码。

请求方式

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

参数	说明
method	方法名称: ch-password
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-password 接口

重置用户密码，无需输入原密码，仅管理员有权限。

请求方式

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

参数	说明
method	方法名称: set-password
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-eth 接口

设置以太网配置信息。

请求方式

```
GET http://ip/usapi?method=set-eth&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-eth
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
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=magewellis-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

新增于: V1.3

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

新增于: V1.3

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

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

新增于: V1.4

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