美乐威 Pro Convert 解码器

前言

通用接口

设备重置

用户登录

设备状态信息

信号状态信息

视频设置

音频设置

EDID 设置

数据源

用户管理

网络设置

固件更新

报告

云平台

系统日志

文档简介

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

本文档中的 API 可适用于以下产品：

- Pro Convert for NDI® to AIO
- Pro Convert for NDI® to HDMI
- Pro Convert for NDI® to HDMI 4K
- Pro Convert for NDI® to SDI

“NDI” 是 NewTek, Inc. 在美国和其他国家的注册商标。

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/mwapi?method=xxx&param1=value1&param2=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 status 属性为 API 状态码，为 0 时表示数据获取或操作成功，否则为相应的失败状态码。

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

API 状态码

```
{
  0: MW_STATUS_SUCCESS,
  1: MW_STATUS_PENDING,
  2: MW_STATUS_TIMEOUT,
  3: MW_STATUS_INTERRUPTED,
  4: MW_STATUS_TRY_AGAIN,
  5: MW_STATUS_NOT_IMPLEMENTED,
  6: MW_STATUS_UNKNOWN_ERROR,
  7: MW_STATUS_INVALID_ARG,
  8: MW_STATUS_NO_MEMORY,
  9: MW_STATUS_UNSUPPORTED,
  10: MW_STATUS_FILE_BUSY,
  11: MW_STATUS_DEVICE_BUSY,
  12: MW_STATUS_DEVICE_LOST,
  13: MW_STATUS_IO_FAILED,
  14: MW_STATUS_READ_FAILED,
  15: MW_STATUS_WRITE_FAILED,
  16: MW_STATUS_NOT_EXIST,
  17: MW_STATUS_TOO_MANY,
  18: MW_STATUS_TOO_LARGE,
  19: MW_STATUS_OVERFLOW,
  20: MW_STATUS_UNDERFLOW,
  21: MW_STATUS_FORMAT_ERROR,
  22: MW_STATUS_FILE_EXISTS,
  23: MW_STATUS_FILE_TYPE_ERROR,
  24: MW_STATUS_DEVICE_TYPE_ERROR,
  25: MW_STATUS_IS_DIRECTORY,
  26: MW_STATUS_READ_ONLY,
  27: MW_STATUS_RANGE_ERROR,
  28: MW_STATUS_BROKEN_PIPE,
  29: MW_STATUS_NO_SPACE,
  30: MW_STATUS_NOT_DIRECTORY,
  31: MW_STATUS_NOT_PERMITTED,
  32: MW_STATUS_BAD_ADDRESS,
  33: MW_STATUS_SEEK_ERROR,
  34: MW_STATUS_CROSS_DEVICE_LINK,
  35: MW_STATUS_NOT_INITIALIED,
  36: MW_STATUS_AUTH_FAILED,
  37: MW_STATUS_NOT_LOGGED_IN,
  38: MW_STATUS_WRONG_STATE,
  39: MW_STATUS_MISMATCH,
  40: MW_STATUS_VERIFY_FAILED,
  41: MW_STATUS_CONSTRAINT_VIOLATION
}
```

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,
}
```

DEMO: 命令行

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

不同操作系统中，下边示例的 cookie 文件存放位置不同，请根据实际情况修改。

wget

1.登录并保存 cookies

```
wget --save-cookies=/var/tmp/sid.txt --keep-session-cookies "http://192.168.66.1/mwapi?method=login&id=Admin&pas
s=e3afed0047b08059d0fada10f400c1e5" -q -O -
```

2.获取用户列表

```
wget --load-cookies=/var/tmp/sid.txt --keep-session-cookies "http://192.168.66.1/mwapi?method=get-users" -q -O -
```

3.新增用户

```
wget --load-cookies=/var/tmp/sid.txt --keep-session-cookies "http://192.168.66.1/mwapi?method=add-user&id=test&p
ass=c4ca4238a0b923820dcc509a6f75849b" -q -O -
```

curl

1.登录并保存 cookies

```
curl --cookie-jar /var/tmp/sid.txt "http://192.168.66.1/mwapi?method=login&id=Admin&pass=e3afed0047b08059d0fada1
0f400c1e5"
```

2.获取用户列表

```
curl --cookie /var/tmp/sid.txt "http://192.168.66.1/mwapi?method=get-users"
```

3.新增用户

```
curl --cookie /var/tmp/sid.txt "http://192.168.66.1/mwapi?method=add-user&id=test&pass=c4ca4238a0b923820dcc509a6
f75849b"
```

DEMO: Node.js

本文通过 Node.js 环境下的两个实例介绍如何调用 Pro Convert Decoder API。

DEMO 下载链接：pro-convert-api-demo-nodejs.zip

DEMO 目录结构：

```
pro-convert-api-demo-nodejs
|
|-- httpUtils.js    // 基于 nodejs 的 HTTP 模块封装了 get 方法和 upload 方法
|-- DEMO_EDID.bin   // upload.js 调用 upload-edid 接口时默认的上传文件，使用时请替换为自己设备的 EDID 文件
|-- 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 pro-convert-api-demo-nodejs
```

2.运行 get.js

```
node get
```

3.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：pro-convert-api-demo-c.zip
- 编译 "pro_convert_curl.c"，链接到"libcurl"
- 生成可执行性文件 "pro_convert_curl"

运行方式

- 在终端控制台进入 bin 目录、执行相应平台的 pro_convert_curl。

```
cd pro-convert-api-demo-c/bin/linux
./pro_convert_curl <hostip:port>
```

- 输出结果

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

***** 2. get caps *****
get caps response data:
{
    "status": 0,
    "max-input-width": 4096,
    "max-input-height": 2160,
    "max-output-width": 4096,
    "max-output-height": 2160,
    "has-input": true,
    "has-output": true,
    "has-loop-through": true,
    "has-fan": true,
    "has-input-edid": true,
    "has-output-edid": true,
    "has-sdcard": true,
    "has-ptz": true
}

***** 3. upload EDID *****
upload EDID response data:
{
    "status": 0,
    "data": "AP///////wA09wEAAQAAAAEaAQOAAAB4Au6Vo1RMmSYPUFT//4AxQEVAYUBxQIGA0QDhwAEACOgAMPJwWoCwWIoAUB10AAAeAjq
AGHE4LUBYLEUAUB10AAAeAAAA/QAPlg+HPAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdXl9gZWZiY2QHFgMSMgl
/BxUHUD0GwFcGAF9/AWd/AINPAADiAA9uAwwAEAC4eCEQgAECAwRn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUQAeMEaPMwBQHXQ
AAB4AAAAAAAAAAAAAzw=="
}
```

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/mwapi?method=ping
```

参数	说明
method	ping

返回数据

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

属性	说明
status	0：设备可以访问。返回其它值请参考 API 状态码。

sync-time 接口

同步设备系统时间，仅管理员有权限。

为了确保系统时间准确，请在管理员登录后自动同步一次。

请求方式

```
GET http://ip/mwapi?method=sync-time&date=xxx&time=xxx
```

参数	说明
method	方法名称：sync-time
date	UTC 日期，格式：dd/MM/yyyy。
time	UTC 时间，格式：HH:mm:ss。

返回数据

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

属性	说明
status	返回状态。0：系统时间同步成功，返回其它值请参考 API 状态码。

接口示例

```
http://192.168.66.1/mwapi?method=sync-time&date=03%2F19%2F2019&time=07:02:26
```

get-caps 接口

不同型号的设备规格有差异，调用此接口可以获取当前连接的设备规格。

请求方式

```
GET http://ip/mwapi?method=get-caps
```

参数	说明
method	方法名称：get-caps

返回数据

```
{
  "status": 0,
  "max-input-width": 1920,
  "max-input-height": 1080,
  "max-output-width": 4096,
  "max-output-height": 2160,
  "has-input": false,
  "has-output": true,
  "has-loop-through": false,
  "has-fan": false,
  "has-input-edid": false,
  "has-output-edid": true,
  "has-sdcard": true,
  "has-ptz": false,
  "has-ndi": true,
  "has-alpha-disp-mode": true
}
```

属性	说明
status	0：数据获取成功。返回其它值请参考 API 状态码。
max-input-width	显示输入信号水平方向的总像素数。
max-input-height	显示输入信号垂直方向的总行数。
max-output-width	设备输出信号水平方向的总像素数。
max-output-height	显示输出信号垂直方向的总行数。
has-input	设备是否有输入接口，有效值：true/false。
has-output	设备是否有输出接口，有效值：true/false。
has-loop-through	设备是否支持环出，有效值：true/false。
has-fan	设备是否有风扇，有效值：true/false。
has-sdcard	设备是否支持 SD 卡功能，有效值：true/false。
has-ptz	设备是否支持 PTZ 功能，有效值：true/false。
has-input-edid	设备是否支持输入端口 EDID，有效值：true/false。
has-output-edid	设备是否支持输出端口 EDID，有效值：true/false。
has-ndi	设备数据源是否支持 NDI 源，有效值：true/false。
has-alpha-disp-mode	设备视频处理是否支持 Alpha 通道，有效值：true/false。

reboot 接口

此接口用于重启设备，重启后需要重新登录，仅管理员有权限。

重启过程需要几分钟时间，可以使用 ping 接口 判断设备是否已经重启。

请求方式

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

参数	说明
method	方法名称：reboot

返回数据

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

属性	说明
status	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。

get-auto-reboot 接口

获取设备自动重启的配置信息。

请求方式

```
GET http://ip/mwapi?method=get-auto-reboot
```

参数	说明
method	方法名称：get-auto-reboot

返回数据

```
{
  "status": 0,
  "enable": true
  "hour": 3
  "min": 30
  "week-flags": 8
}
```

属性	说明
status	0：数据获取成功。返回其它值请参考 API 状态码。
enable	是否启用自动重启功能，有效值：true/false。
week-flags	所选日期的掩码总和，周一到周日的掩码分别为：1、2、4、8、16、32、0。如: 选择了周一和周三，week-flags=1+4=5
hour	时间，24小时制，有效值范围：0 ~ 23。
min	分钟，有效值范围：0 ~ 59。

所设置的时间需要转换成 UTC 时间

set-auto-reboot 接口

设置设备自动重启的配置信息。

请求方式

```
GET http://ip/mwapi?method=set-auto-reboot&enable=true&week-flags=2&hour=12&min=21
```

参数	说明
method	方法名称：set-auto-reboot
enable	是否启用自动重启功能，有效值：true/false。
week-flags	所选日期的掩码总和，周一到周日的掩码分别为：1、2、4、8、16、32、0。如: 选择了周一和周三，week-flags=1+4=5
hour	时间，24小时制，有效值范围：0 ~ 23。
min	分钟，有效值范围：0 ~ 59。

所设置的时间需要转换成 UTC 时间

返回数据

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

属性	说明
status	0：数据获取成功。返回其它值请参考 API 状态码。

get-reset-all-permission 接口

通过该接口来判断是否需要为用户提供 `重置设备` 操作。

设备可以通过两种方式连接以太网：通过连接网线，或通过 USB 线缆连接电脑。但 `重置设备` 操作只能在 Ethernet over USB 连接时使用。

重置设备接口：reset-all-settings。

请求方式

```
GET http://ip/mwapi?method=get-reset-all-permission
```

参数	说明
method	方法名称：get-reset-all-permission

返回数据

```
{
  "status": 0,
  "reset-all-enabled": true
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
reset-all-enabled	是否支持重置功能，有效值：true/false。

reset-all-settings 接口

将设备的全部参数恢复至默认值。

仅在通过 **USB 连接网络** 时可执行此操作。

重置后，设备需要重启，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启。

请求方式

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

参数	说明
method	方法名称：reset-all-settings

返回数据

```
{
  status: 0,
  ip-addr: "192.168.66.1",
  estimated-duration: 120
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
ip-addr	设备访问 IP 地址（重置后访问 IP 可能会改变）。
estimated-duration	预计等待时间，以秒为单位。

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/mwapi?method=login&id=xxx&pass=xxx
```

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

返回数据

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

属性	说明
status	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。

接口示例

```
http://192.168.66.1/mwapi?method=login&id=Admin&pass=e3afed0047b08059d0fada10f400c1e5
```

logout 接口

退出登录，返回到登录界面。

请求方式

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

参数	说明
method	方法名称：logout

返回数据

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

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

get-summary-info 接口

获取设备状态信息，包括设备基本信息、以太网状态、USB RNDIS 状态、NDI 状态。

请求方式

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

参数	说明
method	方法名称：get-summary-info

返回数据

JSON 结构如下：

```
{
  "status": 0,
  "device": {...},
  "ethernet": {...},
  "rndis": {...},
  "ndi": {...}
}
```

1. 返回状态

```
"status": 0
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。

2. 设备基本信息 （device {...}）

```
"device": {
  "name": "Pro Convert",
  "model": "NDI to HDMI",
  "serial-no": "101",
  "hw-revision": "A",
  "fw-version": "1.1.157",
  "up-to-date": true,
  "output-state": "connected",
  "cpu-usage": 60.00,
  "memory-usage": 37.46,
  "core-temp": 69.23,
  "board-id": 0,
  "up-time": 19972,
  "sd-size": 0
}
```

属性	说明
name	产品族名称。
model	产品型号，如 NDI to HDMI。
serial-no	设备序列号。
hw-revision	硬件版本号，有效值：A ~ Z。
fw-version	当前设备固件版本号。
up-to-date	当前设备固件版本是否最新。是则显示 true，否则显示 false。
output-state	输出信号状态，有效值：unconnected、unsupported、active。
cpu-usage	CPU 使用率。
memory-usage	内存使用率。

core-temp	当前设备芯片的温度 (℃)。
board-id	拨码开关编号，有效值：0 ~ F。
up-time	距离设备上次启动的时长 (s)。
sd-size	SD 卡容量 (MB)。

3. 以太网状态信息 （ethernet {...}）

```
"ethernet": {
  "state": "disconnected",
  "mac-addr": "70:B3:D5:75:D2:41",
  "ip-addr": "0.0.0.0",
  "ip-mask": "0.0.0.0",
  "gw-addr": "0.0.0.0",
  "dns-addr": "0.0.0.0",
  "tx-speed-kbps": 0,
  "rx-speed-kbps": 0
}
```

属性	说明
state	以太网连接状态，有效值：down、disconnected、10m、100m、1000m、2500m、5000m、10000m。
mac-addr	MAC 地址。
ip-addr	以太网 IP 地址。
ip-mask	子网掩码地址。
gw-addr	网关地址。
dns-addr	DNS 服务器地址。
tx-speed-kbps	以太网发送速度 (Kbps)。
rx-speed-kbps	以太网接收速度 (Kbps)。

4. USB RNDIS 状态信息 （rndis {...}）

```
"rndis": {
  "state": "high-speed",
  "ip-addr": "192.168.66.1",
  "tx-speed-kbps": 0,
  "rx-speed-kbps": 0
}
```

属性	说明
state	USB 连接状态，有效值：disconnected、full-speed、high-speed、super-speed-5g、super-speed-10g。
ip-addr	USB RNDIS 地址。
tx-speed-kbps	通过 USB RNDIS 发送数据速率 (Kbps)。
rx-speed-kbps	通过 USB RNDIS 接收数据速度 (Kbps)。

5. NDI 状态信息 （ndi {...}）

```
"ndi": {
  "name": "#00 (B401180706020)",
  "connected": true,
  "tally-preview": false,
  "tally-program": false,
  "audio-drop-frames": 0,
  "video-drop-frames": 0,
  "video-bit-rate": 0,
  "audio-bit-rate": 0,
  "video-width": 0,
  "video-height": 0,
  "video-scan": "progressive",
  "video-field-rate": 0.00,
  "audio-num-channels": 0,
  "audio-sample-rate": 0,
  "audio-bit-count": 16,
  "audio-jitter": 26,
```

```
  "video-jitter": 7
}
```

属性	说明
name	视频源名称。
connected	NDI 连接状态，连接则显示 true，否则显示 false。
tally-preview	预览状态，预览中则显示 true，否则显示 false。
tally-program	播出状态，播出中则显示 true，否则显示 false。
audio-drop-frames	前一秒音频丢帧数。
audio-bit-rate	前一秒的音频编码速率 (Kbps)。
audio-num-channels	音频通道总数。
audio-sample-rate	音频采样率，包括 32000、44100 等。
audio-bit-count	音频比特率，包括 16、20、24 等。
video-drop-frames	前一秒的视频丢帧数。
video-bit-rate	前一秒的视频编码速率 (Kbps)。
video-width	视频宽度，总像素数。
video-height	视频高度，总像素数。
video-scan	扫描方式：progressive、interlaced、psf。
video-field-rate	场率，包括 24、25、29.97、30、48、50、59.94、60。
audio-jitter	音频抖动。
video-jitter	视频抖动。

get-signal-info

获取设备输入信号状态信息。

请求方式

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

参数	说明
method	方法名称：get-signal-info

返回数据

JSON 结构如下：

```
{
  "status": 0,
  "signal-info-types": ["video-info"、"audio-info"、"hdmi-info"、"sdi-info"、"info-frames"]、 // 数组中各项和下边属性一一对应
  "video-info": {...},
  "audio-info": {...},
  "hdmi-info": {...},
  "sdi-info": {...},
  "info-frames": {...}
}
```

返回状态

```
"status": 0
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。

信号类型

```
"signal-info-types": [
  "video-info",    // 视频信号状态
  "audio-info",    // 音频信号状态
  "hdmi-info",     // HDMI 信号状态
  "sdi-info",      // SDI 信号状态
  "info-frames"     // 信息帧
]
```

视频信号状态 （video-info：{...}）

```
"video-info": {
  "codec": "uncompressed",
  "width": 1920,
  "height": 1080,
  "scan": "progressive",
  "field-rate": 60.00,
  "color-depth": 8,
  "color-format": "rgb",
  "aspect-ratio": "16:9",
  "sampling": "4:4:4",
  "quant-range": "full",
  "sat-range": "full",
  "frame-struct": "2d"
}
```

属性	说明

codec	编码类型，包括 uncompressed、dsc、mpeg2 等。
width	视频宽度，像素数。
height	视频高度，像素数。
scan	扫描方式，有效值：progressive、interlaced、psf。
field-rate	场率，包括 24、25、29.97、30、48、50、59.94、60。
color-depth	色深，包括 8、10、12。
color-format	色彩空间，有效值：rgb、bt.601、bt.709、bt.2020。
aspect-ratio	显示宽高比，包括 16:9、4:3 等。
sampling	采样方式，包括 4:2:0、4:2:2、4:4:4、4:4:4:4。
quant-range	量化范围，有效值：limited、full。
sat-range	饱和范围，有效值：limited、extended、full。
frame-struct	帧结构，有效值：2d、3d-left-right、3d-top-bottom、3d-left-right-half、3d-top-bottom-half。

音频信号状态 （audio-info：{...}）

```
"audio-info": {
  "codec": "lpcm",
  "num-channels": 2,
  "sample-rate": 48000,
  "bit-count": 16
}
```

属性	说明
codec	编码方式，包括 lpcm、ac3、aac 等。
num-channels	通道数，包括 1、2、.. 16。
sample-rate	采样率，包括 32000、44100 等。
bit-count	比特率，包括 16、20、24 等。

HDMI 信号状态 （hdmi-info：{...}）

```
"hdmi-info": {
  "mode": "dvi",
  "vic": 0,
  "scramble": false,
  "clock-ratio": 1,
  "hdcp": "none",
  "repeat-count": 0,
  "it-content": false,
  "timing-mode-line": "148.50 1920 2008 2052 2200 1080 1084 1089 1125 +hsync +vsync"
}
```

属性	说明
mode	模式，包括 hdmi、dvi。
vic	视频标识码。
scramble	有效值：true/false。
clock-ratio	CPU 倍频，有效值：1、4。
hdcp	HDCP 加密方式，有效值：none、hdcp-1.x、hdcp-2.2。
repeat-count	重复计数，包括 0、1、2、4...
it-content	IT 内容标志，有效值：true/false。
timing-mode-line	modeline 时序格式。 格式：pclk hdisp hsyncstart hsyncend htotal vdisp vsyncstart vsyncend vtotal [flags]。 flags：+hsync、-hsync、+vsync、-vsync、interlace、double-scan、sog、+csync、-csync。 例子：23.86 640 656 720 800 480 481 484 497 -hsync +vsync。 pclk 单位为 MHz，其他单位为像素。

SDI 信号状态 （sdi-info：{...}）

```
"sdi-info": {
```

```
  "link-type": "",
  "link-speed": "",
  "stream-type": "",
  "level-b": true,
  "interlaced": true,
  "assignment": 0,
  "st352-payload-id": 3423424,
  "h-total": 2250,
  "v-total": 1125,
  "h-active": 1920,
  "v-active": 1080
}
```

属性	说明
link-type	链路类型，有效值：unknown、single-link、dual-link、quad-link。
link-speed	传输速率，有效值：unknown、sd、hd、3g、6g、12g。
stream-type	数据流类型，有效值：single-stream、dual-stream、3d。
level-b	Level B，有效值：true/false。
interlaced	隔行信号，有效值：true/false。
assignment	链路编号。
st352-payload-id	ST 352 负载 ID，无符号 32 位整型，转换成 16 进制显示。
h-total	水平总宽度，像素数。
v-total	垂直总高度，像素数。
h-active	水平有效宽度，像素数。
v-active	垂直有效高度，像素数。

信息帧 （info-frames：{...}）

```
"info-frames": [
  {
    "id": "AVI"、
    "type": 130,
    "version": 2,
    "checksum": 96,
    "data": "ACgAIgAAADkEAACBBw=="
  },
  {
    "id": "Audio",
    "type": 132,
    "version": 1,
    "checksum": 112,
    "data": "AQAAAAAAAAAAAA=="
  }
]
```

属性	说明
id	信息帧名称，包括 AVI、Audio...
type	报文类型。
version	版本。
checksum	校验和。
data	base64 编码数据，转换成 16 进制显示。

get-video-config 接口

获取视频设置信息。

请求方式

```
GET http://ip/mwapi?method=get-video-config
```

参数	说明
method	方法名称：get-video-config

返回数据

```
{
  "status": 0,
  "show-title": false,
  "show-tally": false,
  "show-vu-meter": true,
  "vu-meter-mode": "none",
  "show-center-cross": false,
  "safe-area-mode": "none",
  "ident-mode": "none",
  "ident-text": "",
  "h-flip": false,
  "v-flip": false,
  "switch-mode": "blank",
  "deinterlace-mode": "bob",
  "in-auto-color-fmt": true,
  "in-color-fmt": "bt.709",
  "ar-convert-mode": "full",
  "alpha-disp-mode": "alpha-blend-checkerboard",
  "follow-input-mode": true
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
show-title	是否显示视频源名称和分辨率，有效值：true/false。
show-tally	是否显示 Tally 灯状态，有效值：true/false。
show-vu-meter	是否显示音频仪表盘，有效值：true/false。
vu-meter-mode	音频表刻度，有效值：none、db、post-gain-db、post-gain-dbfs
show-center-cross	是否显示中心十字，有效值：true/false。
safe-area-mode	安全区域，有效值：none、4:3、80%、square。
ident-mode	标识模式，有效值：none、ident-text、device-name
ident-text	标识文本，字符要求： 1、1 - 32 个字符 2、仅支持 A-Z, a-z, 0-9 和特殊符号，包括空格、下划线（_）、减号（-）、加号（+）。
h-flip	是否水平翻转，有效值：true/false。
v-flip	是否垂直翻转，有效值：true/false。
switch-mode	视频源切换显示，有效值：keep-last、blank
deinterlace-mode	去隔行模式，有效值：bob、weave
in-auto-color-fmt	色彩空间编码是否设为自动获取，有效值：true/false。
in-color-fmt	色彩空间编码，有效值：bt.601、bt.709
ar-convert-mode	宽高比模式，有效值：windowbox、full、zoom
alpha-disp-mode	Alpha 通道模式，有效值：alpha-only、alpha-blend-white、alpha-blend-black、alpha-blend-checkerboard
follow-input-mode	输出分辨率是否与输入源保持一致，仅适用于产品： NDI to AIO、NDI to HDMI 和 NDI to SDI。有效值：true/false。

get-def-video-config 接口

获取视频默认的配置信息。

请求方式

```
GET http://ip/mwapi?method=get-def-video-config
```

参数	说明
method	方法名称：get-def-video-config

返回数据

```
{
  "status": 0,
  "show-title": false,
  "show-tally": false,
  "show-vu-meter": true,
  "vu-meter-mode": "none",
  "show-center-cross": false,
  "safe-area-mode": "none",
  "ident-mode": "none",
  "ident-text": "",
  "h-flip": false,
  "v-flip": false,
  "switch-mode": "blank",
  "deinterlace-mode": "bob",
  "in-auto-color-fmt": true,
  "in-color-fmt": "bt.709",
  "ar-convert-mode": "full",
  "alpha-disp-mode": "alpha-blend-checkerboard"
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
show-title	是否显示视频源名称和分辨率，有效值：true/false。
show-tally	是否显示 Tally 灯状态，有效值：true/false。
show-vu-meter	是否显示音频仪表盘，有效值：true/false。
vu-meter-mode	音频表刻度，有效值：none、db、post-gain-db、post-gain-dbfs
show-center-cross	是否显示中心十字，有效值：true/false。
safe-area-mode	安全区域，有效值：none、4:3、80%、square。
ident-mode	标识模式，有效值：none、ident-text、device-name
ident-text	标识文本，字符要求： 1、1 - 32 个字符 2、仅支持 A-Z, a-z, 0-9 和特殊符号，包括空格、下划线（_）、减号（-）、加号（+）。
h-flip	是否水平翻转，有效值：true/false。
v-flip	是否垂直翻转，有效值：true/false。
switch-mode	视频源切换显示，有效值：keep-last、blank
deinterlace-mode	去隔行模式，有效值：bob、weave
in-auto-color-fmt	色彩空间编码是否设为自动获取，有效值：true/false。
in-color-fmt	色彩空间编码，有效值：bt.601、bt.709
ar-convert-mode	宽高比模式，有效值：windowbox、full、zoom
alpha-disp-mode	Alpha 通道模式，有效值：alpha-only、alpha-blend-white、alpha-blend-black、alpha-blend-checkerboard

set-video-config 接口

修改视频配置信息。

请求方式

```
GET http://ip/mwapi?method=set-video-config&param1=value1&param2=value2...
```

返回数据

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

属性	说明
status	返回状态。0：设置成功，返回其它值请参考 API 状态码。

接口示例

1. OSD 设置

```
http://ip/mwapi?method=set-video-config&show-title=true&show-tally=true&show-vu-meter=true&vu-meter-mode=dbu&safe-area-mode=4:3&show-center-cross=true
```

参数	说明
method	方法名称：set-video-config
show-title	是否显示视频源名称和分辨率，有效值：true/false。
show-tally	是否显示 Tally 灯状态，有效值：true/false。
show-vu-meter	是否显示音频仪表盘，有效值：true/false。
vu-meter-mode	音频表刻度，有效值：none、db、post-gain-db、post-gain-dbfs
show-center-cross	是否显示中心十字，有效值：true/false。
safe-area-mode	安全区域，有效值：none、4:3、80%、square。
ident-mode	标识模式，有效值：none、ident-text、device-name
ident-text	标识文本，字符要求： 1、1 - 32 个字符 2、仅支持 A-Z, a-z, 0-9 和特殊符号，包括空格、下划线（_）、减号（-）、加号（+）。

2. 处理设置

```
http://ip/mwapi?method=set-video-config&h-flip=false&v-flip=false&deinterlace-mode=weave&ar-convert-mode=full&alpha-disp-mode=alpha-blend-checkerboard
```

参数	说明
method	方法名称：set-video-config
h-flip	是否水平翻转，有效值：true/false。
v-flip	是否垂直翻转，有效值：true/false。
deinterlace-mode	去隔行模式，有效值：bob、weave
ar-convert-mode	宽高比模式，有效值：windowbox、full、zoom
alpha-disp-mode	Alpha 通道模式，有效值：alpha-only、alpha-blend-white、alpha-blend-black、alpha-blend-checkerboard

3. 视频源设置

```
http://ip/mwapi?method=set-video-config&in-auto-color-fmt=false&in-color-fmt=bt.709&switch-mode=blank
```

参数	说明
method	方法名称：set-video-config
in-auto-color-fmt	色彩空间编码是否设为自动获取，有效值：true/false。
in-color-fmt	色彩空间编码，有效值：bt.601、bt.709
switch-mode	视频源切换显示，有效值：keep-last、blank

4. 设置输出分辨率是否与输入源保持一致

```
http://ip/mwapi?method=set-video-config&follow-input-mode=false
```

参数	说明
follow-input-mode	输出分辨率是否与输入源保持一致，仅适用于产品： NDI to AIO、NDI to HDMI 和 NDI to SDI。有效值：true/false。

get-supported-video-modes 接口

获取输出显示器所支持的分辨率列表。

请求方式

```
GET http://ip/mwapi?method=get-supported-video-modes
```

参数	说明
method	方法名称：get-supported-video-modes

返回数据

JSON 结构如下：

```
{
  "status": 0,
  "modes": [...]
}
```

1. 返回状态

```
"status": 0
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。

2. 显示设备支持的分辨率列表信息 （modes {...}）

```
{
  "modes": [
    {
      "width": 2560,
      "height": 1440,
      "interlaced": false,
      "field-rate": 5995,
      "aspect-ratio": 1.77777779,
      "pref-mode": true,
      "curr-mode": true
    },
    {
      "width": 2560,
      "height": 1440,
      "interlaced": false,
      "field-rate": 14391,
      "aspect-ratio": 1.77777779,
      "pref-mode": false,
      "curr-mode": false
    }
    ...
}
```

属性	说明
width	视频宽度，像素数目
height	视频高度，像素数目
interlaced	是否为隔行扫描，有效值：true/false
field-rate	帧率
aspect-ratio	宽高比
pref-mode	是否为推荐值，有效值：true/false

curr-mode	是否为当前应用值，有效值：true/false

set-video-mode 接口

设置显示分辨率。分辨率必须为 get-supported-video-modes 中获取的值。

请求方式

```
GET http://ip/mwapi?method=set-video-mode&param1=value1&param2=value2...
```

返回数据

```
{
  "status": 0,
    "width": 720,
    "height": 576,
    "interlaced": false,
    "field-rate": 5000,
    "aspect-ratio": 1.25000000
}
```

属性	说明
status	返回状态。0：设置成功，返回其它值请参考 API 状态码。
width	视频宽度，像素数目
height	视频高度，像素数目
interlaced	是否为隔行扫描，有效值：true/false
field-rate	帧率
aspect-ratio	宽高比

接口示例

```
http://ip/mwapi?method=set-video-mode&width=720&height=576&aspect-ratio=1.25&field-rate=5000&interlaced=false
```

参数	说明
method	方法名称：set-video-mode
width	视频宽度，像素数目
height	视频高度，像素数目
interlaced	是否为隔行扫描，有效值：true/false
field-rate	帧率
aspect-ratio	宽高比

reset-video-config 接口

将视频所有设置恢复为出厂时的默认值。

请求方式

```
GET http://ip/mwapi?method=reset-video-config
```

参数	说明
method	方法名称：reset-video-config

返回数据

```
{
  "status": 0,
  "show-title": false,
  "show-tally": false,
  "show-vu-meter": true,
  "vu-meter-mode": "none",
  "show-center-cross": false,
  "safe-area-mode": "none",
  "ident-mode": "none",
  "ident-text": "",
  "h-flip": false,
  "v-flip": false,
  "switch-mode": "blank",
  "deinterlace-mode": "bob",
  "in-auto-color-fmt": true,
  "in-color-fmt": "bt.709",
  "ar-convert-mode": "full",
  "alpha-disp-mode": "alpha-blend-checkerboard"
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
show-title	是否显示视频源名称和分辨率，有效值：true/false。
show-tally	是否显示 Tally 灯状态，有效值：true/false。
show-vu-meter	是否显示音频仪表盘，有效值：true/false。
vu-meter-mode	音频表刻度，有效值：none、db、post-gain-db、post-gain-dbfs
show-center-cross	是否显示中心十字，有效值：true/false。
safe-area-mode	安全区域，有效值：none、4:3、80%、square。
ident-mode	标识模式，有效值：none、ident-text、device-name
ident-text	标识文本，字符要求： 1、1 - 32 个字符 2、仅支持 A-Z, a-z, 0-9 和特殊符号，包括空格、下划线（_）、减号（-）、加号（+）。
h-flip	是否水平翻转，有效值：true/false。
v-flip	是否垂直翻转，有效值：true/false。
switch-mode	视频源切换显示，有效值：keep-last、blank
deinterlace-mode	去隔行模式，有效值：bob、weave
in-auto-color-fmt	色彩空间编码是否设为自动获取，有效值：true/false。
in-color-fmt	色彩空间编码，有效值：bt.601、bt.709
ar-convert-mode	宽高比模式，有效值：windowbox、full、zoom
alpha-disp-mode	Alpha 通道模式，有效值：alpha-only、alpha-blend-white、alpha-blend-black、alpha-blend-checkerboard

get-video-format 接口

获取视频格式信息。

请求方式

```
GET http://ip/mwapi?method=get-video-format
```

参数	说明
method	方法名称：get-video-format

返回数据

```
{
  "status": 0,
  "color-format": "rgb",
  "quant-range": "full"
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
color-format	色彩空间，有效值：rgb、yuv444、yuv422。
quant-range	量化范围，color-format=rgb时才生效，有效值：limited、full。

set-video-format 接口

设置视频格式信息。

请求方式

```
GET http://ip/mwapi?method=set-video-format&color-format=rgb&quant-range=full
```

参数	说明
method	方法名称：set-video-format
color-format	色彩空间，有效值：rgb、yuv444、yuv422。
quant-range	量化范围，color-format=rgb时才生效，有效值：limited、full。

返回数据

```
{
  "status": 0,
  "color-format": "rgb",
  "quant-range": "full"
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
color-format	色彩空间，有效值：rgb、yuv444、yuv422。
quant-range	量化范围，color-format=rgb时才生效，有效值：limited、full。

get-hdmi-output 接口

获取解码后是否输出的状态。

请求方式

```
GET http://ip/mwapi?method=get-hdmi-output
```

参数	说明
method	方法名称：get-hdmi-output

返回数据

```
{
  "status": 0,
  "enabled": true,
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
enabled	解码输出是否启用，有效值：true/false。

set-hdmi-output 接口

设置解码后是否输出。

请求方式

```
GET http://ip/mwapi?method=set-hdmi-output&enabled=false
```

参数	说明
method	方法名称：set-hdmi-output
enabled	解码输出是否启用，有效值：true/false。

返回数据

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

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。

get-audio-config 接口

获取音频设置信息。

请求方式

```
GET http://ip/mwapi?method=get-audio-config
```

参数	说明
method	方法名称：get-audio-config

返回数据

```
{
  "status": 0,
  "gain": -44.00,
  "sample-rate": 0,
  "channels": 0,
  "bit-count": 0,
  "ch0": 0,
  "ch1": 1,
  "ch2": 2,
  "ch3": 3,
  "ch4": 4,
  "ch5": 5,
  "ch6": 6,
  "ch7": 7,
  "ch8": 8,
  "ch9": 9,
  "ch10": 10,
  "ch11": 11,
  "ch12": 12,
  "ch13": 13,
  "ch14": 14,
  "ch15": 15,
  "check-pts": true
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
gain	增益，有效值：-100 ~ 20
sample-rate	采样率，有效值：32000、44100、48000、88200、96000
channels	通道数，有效值：0、2、4、8
bit-count	预留
ch0 ~ ch15	分别代表 1 ~ 16 通道与源通道的映射关系，如： ch0=4&ch1=5 表示 通道1 映射 源通道5，通道2 映射 源通道6
check-pts	是否检测音频 PTS，有效值：true/false。

set-audio-config 接口

修改音频配置信息。

请求方式

```
GET http://ip/mwapi?method=set-audio-config&param1=value1&param2=value2...
```

返回数据

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

属性	说明
status	返回状态。0：设置成功，返回其它值请参考 API 状态码。

接口示例

```
http://ip/mwapi?method=set-audio-config&gain=-60&sample-rate=44100&channels=2&ch0=4&ch1=5&check-pts=true
```

参数	说明
method	方法名称：set-video-config
gain	增益，有效值：-100 ~ 20
sample-rate	采样率，有效值：32000、44100、48000、88200、96000
channels	通道数，有效值： 0：与输入一致 2、ch0, ch1 有效 4、ch0 ~ ch3 有效 8、ch0 ~ ch7 有效
ch0 ~ ch15	分别代表 1 ~ 16 通道与源通道的映射关系，有效通道数由 channels 值决定。如： channels=2&ch0=4&ch1=5 表示使用两声道， 通道1 映射 源通道5，通道2 映射 源通道6
check-pts	是否检测音频 PTS，有效值：true/false。

get-output-edid 接口

获取输出端口 EDID 信息。

请求方式

```
GET http://ip/mwapi?method=get-output-edid
```

参数	说明
method	方法名称：get-output-edid

返回数据


```
{
  "status": 0,
  "data": "AP///////wA09wEAAQAAAAEaAQOAAAB4Au6Vo1RMmSYPUFT//4AxQEVAYUBxQIGA0QDhwAEACOgAMPJwWoCwWIoAUB10AAAeAjqAG
HE4LUBYLEUAUB10AAAeAAAA/QAPlg+HPAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdXl9gZWZiY2QHFgMSMgl/B
xUHUD0GwFcGAF9/AWd/AINPAADiAA9uAwwAEAC4eCEQgAECAwRn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUQAeMEaPMwBQHXQAA
B4AAAAAAAAAAAAAzw=="
}
```


属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
data	EDID 信息，base64格式，显示时需要转换成 16 进制值。

export-edid 接口

导出 EDID，导出格式为.bin。

请求方式

```
GET http://ip/mwapi?method=export-edid&port=out&file-name=xxx.bin
```

参数	说明
method	方法名称：export-edid
port	端口类型，有效值：out。
file-name	导出文件名，文件后缀必须为.bin。

返回数据

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

属性	说明
status	返回状态。0：设置成功，返回其它值请参考 API 状态码。

示例

```
http://ip/mwapi?method=export-edid&port=out&file-name=Output_Port_EDID_2019_03_20_17_36_42.bin
```

list-channels 接口

设备解码数据源主要以下两种：

1. 用户预设的数据源，可以通过 list-channels 接口获取；
2. 设备自动发现的 NDI 数据源，可以通过 get-ndi-sources 接口获取。

请求方式

```
GET http://ip/mwapi?method=list-channels
```

参数	说明
method	方法名称：list-channels

返回数据

```
{
  "status": 0,
  "channels": [
    {
      "name": "RTP",
      "url": "rtp://224.1.2.3:4000?mw-buffer-duration=60"
    },
    {
      "name": "RTP1",
      "url": "rtp://224.2.4.6:6688?mw-buffer-duration=50"
    },
    {
      "name": "UDP",
      "url": "udp://224.1.2.3:4000?mw-buffer-duration=200"
    }
  ]
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
channels	预设数据源数组。

get-channel 接口

获取当前正在使用的数据源信息。

请求方式

```
GET http://ip/mwapi?method=get-channel
```

参数	说明
method	方法名称：get-channel

返回数据

```
{
    "status": 0,
    "name": "5004",
    "ndi-name": false
}
```

字段	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
name	数据源名称
ndi-name	是否为 NDI 数据源，有效值：true/false。

set-channel 接口

选择指定数据源为解码数据源。

请求方式

```
GET http://ip/mwapi?method=set-channel&ndi-name=true&name=xxx
```

参数	说明
method	方法名称：set-channel
ndi-name	是否为 NDI 数据源，有效值：true/false
name	数据源名称

返回数据

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

字段	说明
status	返回状态。0：设置成功，返回其它值请参考 API 状态码。

add-channel 接口

添加预设数据源，目前支持的协议类型有：ntkndi、rtsp、http、rtmp、udp、srt、rtp

请求方式

```
GET http://ip/mwapi?method=add-channel&name=xxx&url=xxx
```

参数	说明
method	方法名称：add-channel
name	数据源名称，唯一值，不能重复 字符要求：1-120 个英文字符
url	数据源 URL

不同协议的**URL**组成参数不同，目前支持的协议有：

1. NTKNDI

```
ntkndi://ndi?name=test&url=192.168.1.90%3A5963&mw-buffer-duration=60
```

URL 组成	说明
ntkndi	NID 数据源协议
ndi	默认主机名，不可修改
name	NDI 流名称
url	NDI 流 URL
mw-buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

2. RTSP

```
rtsp://192.168.1.58:899?mw-buffer-duration=60
```

URL 组成	说明
rtsp://192.168.1.58:899	合法 URL
mw-buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

3. HTTP

```
http://192.168.1.88:8585?mw-buffer-duration=60
```

URL 组成	说明
`http://192.168.1.88:8585`	合法 URL
mw-buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

4. RTMP

```
// RTMP 拉流
rtmp://url/stream-key?mw-buffer-duration=60

// RTMP 推流
rtmp://127.0.0.1/live/stream-key?mw-buffer-duration=80
```

URL 组成	说明
url	RTMP 拉流：合法的 RTMP URL RTMP 推流：rtmp://127.0.0.1/live/.
stream-key	密钥，必须是 URL 最后一个路径，不支持斜杠(/)
mw-buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

5. MPEG-TS over UDP

```
// 组播
udp://ip:port?mw-audio-track=2&mw-buffer-duration=80

// 单播
udp://0.0.0.0:port?mw-audio-track=2mw-buffer-duration=80
```

URL 组成	说明
ip	单播：0.0.0.0 组播：有效 IP 范围 224.0.0.0 ~ 239.255.255.255
port	端口号，有效范围：1 ~ 65535
mw-audio-track	音轨，有效范围：1 ~ 8
mw-buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

6. MPEG-TS over SRT

```
// Caller 模式
srt://ip:port?mode=caller&latency=125&streamid=test&passphrase=1234567890&mw-audio-track=2&mw-buffer-duration=80

// Listener 模式
srt://0.0.0.0:port?mode=listener&latency=125&streamid=test&mw-audio-track=2&mw-buffer-duration=80
```

URL 组成	说明
ip	Listener：0.0.0.0 caller：合法 IP 地址（不能为：0.0.0.0）
port	端口号，有效范围：1 ~ 65535
mode	模式，有效值：caller/listener
latency	延迟时间,有效范围：20 ~ 8000
passphrase	加密密码，选填，需要加密时才设置
streamid	流 ID，字符长度：0 ~ 512
mw-audio-track	音轨，有效范围：1 ~ 8
mw-buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

7. MPEG-TS over RTP

```
// 组播
rtp://ip:port?mw-ts-progid=2&mw-audio-track=2&mw-buffer-duration=80

// 单播
rtp://0.0.0.0:port?mw-ts-progid=2&mw-audio-track=2&mw-buffer-duration=80
```

URL 组成	说明
ip	单播：0.0.0.0 组播：有效 IP 范围 224.0.0.0 ~ 239.255.255.255
port	端口号，有效范围：1 ~ 65535
mw-ts-progid	TS ProgID，有效范围：1 ~ 10000000
mw-audio-track	音轨，有效范围：1 ~ 8
mw-buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

8. TVU ISSP

```
issp://192.168.1.88?mw-buffer-duration=60
```

URL 组成	说明
`issp://192.168.1.88`	合法 URL
mw-buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

返回数据

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

字段	说明
status	返回状态。0：数据添加成功，返回其它值请参考 API 状态码。

modify-channel 接口

修改预设数据源信息。

请求方式

```
GET http://ip/mwapi?method=modify-channel&name=xxx&new-name=xxx&url=xxx
```

参数	说明
method	方法名称：modify-channel
name	指定数据源名称，该值在列表中唯一且不能重复。 字符要求：1-120 个英文字符
new-name	指定数据源备注名，该值在列表中唯一且不能重复。 字符要求：1-120 个英文字符。
url	数据源 URL

返回数据

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

字段	说明
status	返回状态。0：数据修改成功，返回其它值请参考 API 状态码。

del-channel 接口

删除预设列表中的数据源。

请求方式

```
GET http://ip/mwapi?method=del-channel&name=xxx
```

参数	说明
method	方法名称：del-channel
name	待删除的数据源名称

返回数据

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

字段	说明
status	返回状态。0：数据删除成功，返回其它值请参考 API 状态码。

clear-channels 接口

清空所有预设的数据源。

请求方式

```
GET http://ip/mwapi?method=clear-channels
```

参数	说明
method	方法名称：clear-channels

返回数据

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

字段	说明
status	返回状态。0：数据删除成功，返回其它值请参考 API 状态码。

get-buffer-limit 接口

根据数据源协议类型获取各自的缓冲时间信息，包括默认时间值和可调整范围，单位为 ms

请求方式

```
GET http://ip/mwapi?method=get-buffer-limit&proto=ntkndi
```

参数	说明
method	方法名称：get-buffer-limit
proto	协议类型，有效值：ntkndi、rtsp、http、rtmp、udp、srt、rtp

返回数据

```
{
  "status": 0,
  "buffer-duration-def": 60,
  "buffer-duration-min": 20,
  "buffer-duration-max": 120
}
```

字段	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
buffer-duration-def	默认缓冲时间（ms）
buffer-duration-min	最小值（ms）
buffer-duration-max	最大值（ms）

get-ndi-config 接口

获取 NDI 配置信息。

请求方式

```
GET http://ip/mwapi?method=get-ndi-config
```

参数	说明
method	方法名称：get-ndi-config

返回数据

```
{
  "status": 0,
  "enable-discovery": false,
  "discovery-server": "",
  "source-name": "PRO CONVERT (#15 (B410190104001))",
  "group-name": "public",
  "low-bandwidth": false,
  "enable-mcast": true,
  "enable-rudp": false,
  "enable-tcp": false,
  "enable-udp": false,
  "ignore-ndi-hx-video-pts": true
}
```

字段	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
enable-discovery	发现服务开启状态，有效值：true/false。
discovery-server	服务器 IP 地址，多个 IP 地址可用英文逗号（,）隔开
source-name	数据源名称
group-name	客户端群组名称。
low-bandwidth	低带宽开启状态，有效值：true/false。
enable-mcast	UDP（组播）开启状态，有效值：true/false。
enable-rudp	RUDP（单播）开启状态，有效值：true/false。
enable-tcp	TCP（多连接）开启状态，有效值：true/false。
enable-udp	UDP（单播）开启状态，有效值：true/false。
ignore-ndi-hx-video-pts	是否开启忽略 NDI 视频 PTS，有效值：true/false。

set-ndi-config 接口

设置 NDI 配置信息。

请求方式

```
GET http://ip/mwapi?method=set-ndi-config&param1=value1&param2=value2...
```

参数	说明
method	方法名称：set-ndi-config
enable-discovery	发现服务开启状态，有效值：true/false
discovery-server	服务器 IP 地址， enable-discovery = true 时必填
source-name	数据源名称，通过 get-ndi-sources 接口获取
group-name	客户端群组名称，默认 public
low-bandwidth	低带宽开启状态，有效值：true/false
enable-mcast	UDP（组播）开启状态，有效值：true/false。
enable-rudp	RUDP（单播）开启状态，有效值：true/false。
enable-tcp	TCP（多连接）开启状态，有效值：true/false。
enable-udp	UDP（单播）开启状态，有效值：true/false。
ignore-ndi-hx-video-pts	是否开启忽略 NDI 视频 PTS，有效值：true/false。

enable-mcast、enable-rudp、enable-tcp 和 enable-udp 只能一个为 true，但全部为 false 时，表示传输模式为：TCP（单连接）

返回数据

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

属性	说明
status	返回状态。0：设置成功，返回其它值请参考 API 状态码。

get-ndi-sources 接口

获取当前可用的数据源列表。

请求方式

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

参数	说明
method	方法名称：get-ndi-sources

返回数据

```
{
  "status": 0,
  "sources": [
    {
      "ndi-name": "MAGEWELL (USB Capture HDMI (D206191017871))",
      "ip-addr": "192.168.1.192:5963"  // 含有 'amc_id' 的为多播方式
    },
    {
      "ndi-name": "MAGEWELL (USB Capture HDMI (D206191017889))",
      "ip-addr": "192.168.1.192:5961"
    }
  ]
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
sources	当前可用的备用通道数组。

get-playback-config 接口

自动发现的 NDI 数据源的播放配置信息。

请求方式

```
GET http://ip/mwapi?method=get-playback-config
```

参数	说明
method	方法名称：get-playback-config

返回数据

```
{
  "status": 0,
    "buffer-duration": 60
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

set-playback-config 接口

修改播放配置信息。

请求方式

```
GET http://ip/mwapi?method=set-playback-config&buffer-duration=70
```

参数	说明
method	方法名称：set-playback-config
buffer-duration	缓冲时间（ms），有效值可通过 get-buffer-limit 接口获取

返回数据

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

属性	说明
status	返回状态。0：设置成功，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/mwapi?method=get-users
```

参数	说明
method	方法名称：get-users

返回数据

```
{
  "status": 0,
  "users": [
    {
      "id": "Admin",
      "group": "Admin"
    },
    {
      "id": "Test",
      "group": "User"
    }
  ]
}
```

属性	说明
status	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，group：用户组。

add-user 接口

添加用户，仅管理员有权限。

请求方式

```
GET http://ip/mwapi?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，使用 MD5 加密。

返回数据

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

属性	说明
status	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/mwapi?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

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

属性	说明
status	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/mwapi?method=ch-password&pass=xxx&new-pass=xxx
```

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

返回数据

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

属性	说明
status	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/mwapi?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，使用 MD5 加密。

返回数据

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

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

get-eth-status 接口

获取以太网配置信息，仅管理员有权限。

请求方式

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

参数	说明
method	方法名称：get-eth-status

返回数据

```
{
  "status": 0,
  "use-dhcp": true,
  "device-name": "Pro Convert",
  "state": "1000m",
  "mac-addr": "70:B3:D5:75:D2:41",
  "ip-addr": "192.168.1.90",
  "ip-mask": "255.255.255.0",
  "gw-addr": "192.168.1.1",
  "dns-addr": "10.0.0.3",
  "tx-speed-kbps": 0,
  "rx-speed-kbps": 5
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
use-dhcp	是否动态获取 IP 地址，有效值：true/false。
device-name	设备名称。
state	连接状态，有效值：down、disconnected、10m、100m、1000m、2500m、5000m、10000m。
mac-addr	Mac 地址。
ip-addr	IP 地址。
ip-mask	子网掩码。
gw-addr	网关地址。
dns-addr	DNS 服务器地址。
tx-speed-kbps	发送速度 (Kbps)。
rx-speed-kbps	接收速度 (Kbps)。

set-eth-config 接口

设置以太网配置信息。

请求方式

```
GET http://ip/mwapi?method=set-eth-config&param1=value1&param2=value2...
```

参数	说明
method	方法名称：set-eth-config
name	设备名称。
dhcp	是否动态获取 IP 地址，有效值：true/false。
addr	IP 地址。
mask	子网掩码。
gw-addr	网关地址。
dns-addr	DNS 服务器。

返回数据

```
{
  "status": 0,
  "reconnect": true
}
```

属性	说明
status	返回状态。0：设置成功，返回其它值请参考 API 状态码。
reconnect	设置后是否需要重新连接和登录，有效值：true/false。

get-rndis-status 接口

获取 USB RNDIS 配置信息，仅管理员有权限。

请求方式

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

参数	说明
method	方法名称：get-rndis-status。

返回数据

```
{
  "status": 0,
  "state": "high-speed",
  "device-name": "Pro Convert",
  "ip-addr": "192.168.66.1",
  "tx-speed-kbps": 0,
  "rx-speed-kbps": 0
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
state	连接状态，有效值：disconnected、full-speed、high-speed、super-speed-5g、super-speed-10g。
device-name	设备名称。
ip-addr	IP 地址。
tx-speed-kbps	发送速度 (Kbps)。
rx-speed-kbps	接收速度 (Kbps)。

set-rndis-config 接口

设置 USB RNDIS 配置信息。

请求方式

```
GET http://ip/mwapi?method=set-rndis-config&addr=xxx&name=xxx
```

参数	说明
method	方法名称：set-rndis-config
addr	IP 地址，格式为 192.168.xxx.1
name	设备名称

返回数据

```
{
  "status": 0,
  "reconnect": true
}
```

属性	说明
status	返回状态。0：设置成功，返回其它值请参考 API 状态码。
reconnect	修改后是否需要重新连接和登录，有效值：true/false。

get-net-access 接口

获取网络服务配置信息，仅管理员有权限。

请求方式

```
GET http://ip/mwapi?method=get-net-access
```

参数	说明
method	方法名称：get-net-access

返回数据

```
{
  "status": 0,
  "use-ssdp": true,
  "use-https": false,
  "ssl-cert-present": true,
  "ssl-cert-key-present": true
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
use-ssdp	是否启用 SSDP 服务，有效值：true/false。
use-https	是否启用 HTTPS，有效值：true/false。
ssl-cert-present	是否已经上传 CA 证书，有效值：true/false。
ssl-cert-key-present	是否已经上传 CA 私钥，有效值：true/false。

set-net-access 接口

设置网络服务配置信息，仅管理员有权限。

请求方式

```
GET http://ip/mwapi?method=set-net-access&use-ssdp=true&use-https=false
```

参数	说明
method	方法名称：set-net-access
use-ssdp	是否启用 SSDP 服务，有效值：true/false。
use-https	是否启用 HTTPS，有效值：true/false。

返回数据

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

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。

upload-ssl-cert 接口

上传 CA 证书。

请求方式

```
POST http://ip/mwapi?method=upload-ssl-cert
```

参数	说明
method	方法名称：upload-ssl-cert

返回数据

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

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。

upload-ssl-cert-key 接口

上传 CA 私钥。

请求方式

```
POST http://ip/mwapi?method=upload-ssl-cert-key
```

参数	说明
method	方法名称：upload-ssl-cert-key

返回数据

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

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。

get-update-state 接口

获取当前固件版本信息和固件更新状态，仅管理员有权限。

请求方式

```
GET http://ip/mwapi?method=get-update-state
```

参数	说明
method	方法名称：get-update-state

返回数据

idle 状态时的返回值

```
{
  "status": 0,
  "state": "idle",
  "cur-ver": "1.1.72",
  "fw-valid": true
}
```

updating 状态时的返回值

```
{
  "status": 0,
  "state": "updating",
  "cur-ver": "1.1.72",
  "update-to-ver": "1.1.72",
  "num-steps": 4,
  "step-id": 2,
  "step-name": "Erasing image",
  "step-percent": 28,
  "fw-valid": true
}
```

failed 状态时的返回值

```
{
  "status": 0,
  "state": "failed",
  "cur-ver": "1.1.72",
  "error-status": 16,
  "fw-valid": true
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
state	更新状态，有效值：idle、updating、completed、failed。
cur-ver	当前固件版本号。
update-to-ver	目标固件版本号。
num-steps	固件更新步骤总数，仅在 updating 状态下存在。
step-id	正在更新步骤编号，仅在 updating 状态下存在。
step-name	正在更新步骤名称，仅在 updating 状态下存在。
step-percent	当前更新进度，仅在 updating 状态下存在。
error-status	错误状态码，仅在 failed 状态下存在。
fw-valid	当前固件存储是否已损坏，有效值： 1、true：正常 2、false：已损坏。

upload-update-file 接口

上传固件，上传文件格式必须为.mwf。

请求方式

```
POST http://ip/mwapi?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	当前版本是否为最新，有效值：true/false。
version	上传的固件版本号。
size	上传的固件文件大小 (B)。

update 接口

执行更新操作，更新过程中可以通过 get-update-state 接口获取当前状态。

请求方式

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

参数	说明
method	方法名称：update
mode	更新方式，有效值：manual（手动更新）。

返回数据

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

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

get-report 接口

获取设备当前的状态报告，仅管理员有权限。

请求方式

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

返回数据

纯文本格式，html 片段如下。

```
<div class="report-summary">
  <h1>Pro Convert NDI to HDMI</h1>
  <p>Generated at Thu, 21 Mar 2019 07:42:56 GMT</p>
</div>
<div class="report-content">
  <div class="content-level1">
      .
      .
      .
      .
  </div>
</div>
```

export-report 接口

导出设备当前的状态报告，导出文件为 html 格式，仅管理员有权限。

请求方式

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

参数	说明
method	方法名称：export-report
file-name	导出文件名称。

请求结果

直接下载 html 格式报告文件到本地。

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

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/mwapi?method=get-logs&types=xxx,xxx,xxx
```

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

```
{
  "status": 0,
  "logs": [
    {
      "type": "warn",
      "time": "2019-03-19 09:53:03.047",
      "message": "USB state: disconnected"
    },
    {
      "type": "warn",
      "time": "2019-03-19 09:14:09.292",
      "message": "User 'Admin' (192.168.66.2) session 4 timeout"
    },
    {
      "type": "warn",
      "time": "1970-01-01 00:00:11.872",
      "message": "USB state: disconnected"
    },
    ...
  ]
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 API 状态码。
logs	日志数组列表：type：日志类型；time：日志产生时间，message：日志内容。

接口示例

获取全部日志

```
http://192.168.66.1/mwapi?method=get-ptz-configmethod=get-logs&types=all
```

获取 warn 和 error 类型的日志

```
http://192.168.66.1/mwapi?method=get-ptz-configmethod=get-logs&types=warn,error
```

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/mwapi?method=export-logs&file-name=xxx.html
```

参数	说明
method	方法名称：export-logs
file-name	导出的文件名称。

请求结果

直接下载 html 格式日志文件到本地。

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/mwapi?method=clear-logs
```

返回数据

属性	说明
status	返回状态。0：清除成功，返回其它值请参考 API 状态码。