

# USB Fusion

## 前言

文档简介	1.1
接口约定	1.2
API 状态码	1.3
DEMO: 命令行	1.4

## 通用接口

ping	2.1
reboot	2.2
factory-reset-permission	2.3
factory-reset	2.4

## 设备状态信息

get-summary-info	3.1
set-hdmi-output-mode	3.2

## 背景音乐

add-bgm-to-list	4.1
bgm-next	4.2
bgm-play-control	4.3
bgm-previous	4.4
bgm-seek	4.5
bgm-set-policy	4.6
get-bgm-list	4.7
get-bgm-status	4.8
remove-bgm-from-list	4.9
sort-bgm-list	4.10
switch-bgm	4.11

## Switching

create-scene-with-source	5.1
delete-scene	5.2
get-device-status	5.3
get-program-preview	5.4
get-scenes-list	5.5
get-source-thumbnail	5.6
get-video-player-status	5.7
rename-scene	5.8
seek-video-player	5.9
set-switch-action	5.10
set-video-player-status	5.11
switch-scene	5.12
upload-source-file	5.13

## APP 设置

<a href="#">get-app-settings</a>	6.1
<a href="#">get-encoder-params</a>	6.2
<a href="#">set-app-password</a>	6.3
<a href="#">set-encoder-format</a>	6.4
<a href="#">set-network-port</a>	6.5

## 系统

<a href="#">device-info</a>	7.1
<a href="#">info</a>	7.2
<a href="#">set-device-name</a>	7.3
<a href="#">timezone-set</a>	7.4
<a href="#">set-date-time</a>	7.5

## 用户

<a href="#">login</a>	8.1
<a href="#">logout</a>	8.2
<a href="#">get-all</a>	8.3
<a href="#">add</a>	8.4
<a href="#">del</a>	8.5
<a href="#">ch-password</a>	8.6
<a href="#">set-password</a>	8.7

## 网络设置

<a href="#">if-info</a>	9.1
<a href="#">if-set</a>	9.2
<a href="#">if-route</a>	9.3
<a href="#">get-dns</a>	9.4
<a href="#">set-dns</a>	9.5
<a href="#">usb-config</a>	9.6
<a href="#">wireless-mode</a>	9.7
<a href="#">wireless-state</a>	9.8
<a href="#">ap-passwd</a>	9.9
<a href="#">ap-config</a>	9.10
<a href="#">ap-get-config</a>	9.11
<a href="#">wifi-scan</a>	9.12
<a href="#">wifi-scan-results</a>	9.13
<a href="#">wifi-connect</a>	9.14
<a href="#">wifi-connect-auto</a>	9.15
<a href="#">wifi-connect-state</a>	9.16
<a href="#">wifi-disconnect</a>	9.17
<a href="#">wifi-forget</a>	9.18

## 固件更新

<a href="#">online-check</a>	10.1
<a href="#">online-check-result</a>	10.2
<a href="#">upload-fw</a>	10.3

update	10.4
state	10.5
clear	10.6

## 系统日志

clear	11.1
filter	11.2
export	11.3

## WebSocket

MSG_WEB_SORKET_EVENT_CHANGE_INFO	12.1
----------------------------------	------

## 相册

get-album-files-list	13.1
remove-album-file	13.2

## 音频

get-volume-config	14.1
set-volume-config	14.2
get-audio-card-list	14.3
get-audio-card-mixer	14.4
get-mic-gain	14.5
get-mic-monitor	14.6
get-usb-audio-list	14.7
set-audio-card-mixer	14.8
set-mic-gain	14.9
set-mic-monitor	14.10
set-usb-audio	14.11

## 通用设置

add-launch-scene	15.1
get-server-launch-scenes	15.2
get-server-settings	15.3
set-auto-switch	15.4
set-launch-scene	15.5
set-usb-mirror	15.6

## 输入

export-edid	16.1
get-edid-config	16.2
set-default-edid	16.3
set-edid-config	16.4
upload-edid	16.5
get-signal-info	16.6
get-video-config	16.7
set-video-config	16.8

<a href="#">get-webcam-config</a>	16.9
<a href="#">set-webcam-config</a>	16.10

## 笔记

<a href="#">get-note-thumbnail</a>	17.1
------------------------------------	------

## 输出

<a href="#">export-output-edid</a>	18.1
<a href="#">get-hdmi-output-info</a>	18.2
<a href="#">set-hdmi-output-mode</a>	18.3
<a href="#">get-usb-output-config</a>	18.4

## 演示稿

<a href="#">add-presentation</a>	19.1
<a href="#">delete-presentation</a>	19.2
<a href="#">get-presentations-list</a>	19.3
<a href="#">rename-presentation</a>	19.4
<a href="#">switch-presentation</a>	19.5

## 资源

<a href="#">check-source-exist</a>	20.1
<a href="#">check-source-used</a>	20.2
<a href="#">delete-source</a>	20.3
<a href="#">get-sources-list</a>	20.4
<a href="#">import-source</a>	20.5
<a href="#">rename-source</a>	20.6
<a href="#">upload-source-file</a>	20.7

## 文档简介

针对USB Fusion，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：

- USB Fusion

# 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 和 PUT 方式, 文件上传用 POST 方式
- 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 status 属性为 [API 状态码](#), 为 0 时表示数据获取或操作成功, 否则为相应的失败状态码。

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

## API 状态码

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

## DEMO: 命令行

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

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

### wget

#### 1.登录并保存 cookies

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

#### 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=c4ca4238a0b923820dcc509a6f75849b"
```



## ping 接口

判断设备是否可以访问，无需登录。

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

### 请求方式

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

### 返回数据

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

属性	说明
status	0: 设备可以访问。返回其它值请参考 <a href="#">API 状态码</a> 。

## reboot 接口

此接口用于重启设备，重启后需要重新登录，仅管理员有权限。

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

### 请求方式

```
GET/POST /api/reboot
```

### 返回数据

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

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 <a href="#">API 状态码</a> 。

## factory-reset-permission 接口

通过该接口判断是否允许重置设备, 无需登录。

### 请求方式

GET/POST /api/factory-reset-permission

### 返回数据

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

属性	说明
status	0: 设备可以访问。返回其它值请参考 <a href="#">API 状态码</a> 。

## factory-reset 接口

恢复出厂设置。

### 请求方式

GET/POST /api/factory-reset

### 返回数据

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

属性	说明
status	0: 设备可以访问。返回其它值请参考 <a href="#">API 状态码</a> 。

# 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 Net状态。

## 请求方式

```
POST http://ip/mwapi/get-summary-info
```

## 返回数据

JSON 结构如下：

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

### 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 <a href="#">API 状态码</a> 。

### 2. 设备基本信息 (device {...})

```
"device": {
  "name": "USB Presenter",
  "model": "HDMI 2K Plus",
  "serial-no": "B401180706020",
  "hw-revision": "B",
  "fw-version": "1.1.72",
  "up-to-date": true,
  "input-hdmi-1-state": "1920x1080p60",
  "input-hdmi-2-state": "no-signal",
  "input-web-camera-state": "1280x720p30",
  "hdmi-output-mode": 0,
  "hdmi-output-state": "unconnected",
  "cpu-usage": 5.00,
  "memory-usage": 58.33,
  "core-temp": 46.76,
  "up-time": 8006,
  "storage-size": 0,
  "fan-rpm": 0
}
```

属性	说明
name	产品族名称。
model	产品型号，如 HDMI 2K Plus...
serial-no	设备序列号。
hw-version	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。
input-hdmi-1-state	HDMI 1 接口输入信号状态，有效值：no-signal、locking、unsupported 或实际值，如：1920x1080p60。
input-hdmi-2-state	HDMI 2 接口输入信号状态，有效值：no-signal、locking、unsupported 或实际值，如：1920x1080p60。

input-web-camera-state	WEB CAMERA 接口输入信号状态, 有效值: no-signal、unsupported 或实际值, 如: 1920x1080p60。
hdmi-out-mode	输出信号模式, 有效值: 0、1、2, 分别代表 loopthru HDMI 1, loopthru HDMI 2, pgm。
hdmi-out-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 (°C)。
up-time	距离设备上上次启动的时长 (s)。
storage-size	存储容量 (MB)。
fan-rpm	当前每分钟风扇转速。

### 3. 无线网络状态信息 (wifi {...})

```
"wifi": {
  "state": "magemwell-001",
  "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	WiFi连接状态, 有效值: down、disconnected。
mac-addr	MAC 地址。
ip-addr	Wifi 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	Wifi发送速度 (Kbps)。
rx-speed-kbps	Wifi接收速度 (Kbps)。

### 4. 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)。

### 5. 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)。

## 6. USB OUT 状态信息 (out {...})

```
"out": {
  "video-format": "NV12",
  "connection": "USB 3.0",
  "audio-drop-frames": 0,
  "video-drop-frames": 0,
  "video-width": 1920,
  "video-height": 1080,
  "video-scan": "progressive",
  "video-field-rate": 0.00,
  "audio-num-channels": 2,
  "audio-sample-rate": 48000,
  "audio-bit-count": 16
}
```

属性	说明
video-format	UVC 视频输出格式, 可能为 NV12, YUYV, MJPEG 等。
connection	usb连接状态, "USB 3.0", "USB 2.0", "disable"。
audio-drop-frames	前一秒音频丢帧数。
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	帧率, 包括 24、25、29.97、30、48、50、59.94、60。

## set-hdmi-output-mode 接口

设置hdmi 输出模式。

### 请求方式

```
POST http://ip/mwapi/set-hdmi-output-mode
```

参数	说明
mode	mode 类型, 0: LOOPTHRU HDMI 1, 1: LOOPTHRU HDMI 2, 2: PGM。

### 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 <a href="#">API 状态码</a> 。



# add-bgm-to-list接口

添加音频文件至当前列表

## 请求方式

POST http://ip/mwapi/add-bgm-to-list

参数	说明
id	资源ID

```
{  
  "id":203  
}
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

## bgm-next 接口

切换到下一首

### 请求方式

```
POST http://ip/mwapi/bgm-next
```

### 返回数据

JSON 结构如下:

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

#### 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 <a href="#">API 状态码</a> 。

# bgm-play-control 接口

背景音乐播放控制

## 请求方式

```
POST http://ip/mwapi/bgm-play-control
```

参数	说明
playStatus	播放状态, 0:暂停 1:播放

```
{  
  "playStatus":0  
}
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

## bgm-previous 接口

切换到上一首

### 请求方式

```
POST http://ip/mwapi/bgm-previous
```

### 返回数据

JSON 结构如下:

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

#### 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 <a href="#">API 状态码</a> 。

# bgm-seek 接口

背景音乐seek到指定进度

## 请求方式

POST http://ip/mwapi/bgm-seek

参数	说明
progress	seek锚点:0~1000000

```
{  
  "progress": 500000  
}
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

# bgm-set-policy 接口

设置BGM播放策略

## 请求方式

```
POST http://ip/mwapi?method=bgm-set-policy
```

参数	说明
musicPolicy	播放策略, 1:顺序播放 2:随机播放 3:单曲循环 4:整体循环

```
{  
  "musicPolicy":1  
}
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

# get-bgm-list 接口

获取bgm列表

## 请求方式

POST http://ip/mwapi/get-bgm-list

## 返回数据

JSON 结构如下:

```
{
  "info": {
    "musicList": [{...}],
    "pageIndex": 0,
    "totalCount": 3
  },
  "message": "success",
  "result": 0
}
```

### 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 <a href="#">API 状态码</a> 。
pageIndex	分页页码: 0,1,2...
totalCount	items总数

### 2. 当前分页背景音乐列表 bgmList {...}

```
"bgmList": [
  {
    "artist": "Magewell",
    "duration": 231000,
    "id": 2006,
    "title": "Magewell"
  },
  {
    "artist": "Magewell",
    "duration": 194000,
    "id": 2038,
    "title": "Magewell 02"
  },
  {
    "artist": "Magewell",
    "duration": 240000,
    "id": 2037,
    "title": "Magewell 03"
  }
]
```

属性	说明
id	背景音乐ID。
title	音乐名称
artist	演唱者/演奏者
duration	时长 (单位:毫秒)





# get-bgm-status 接口

获取bgm播放状态

## 请求方式

```
POST http://ip/mwapi/get-bgm-status
```

## 返回数据

JSON 结构如下:

```
{
  "info": {},
  "message": "success",
  "result": 0
}
```

### 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 <a href="#">API 状态码</a> 。
musicStatus	背景音乐状态

### 2. 当前状态

```
"musicStatus": {
  "currentMusicId": 2038,
  "duration": 194000,
  "policy": 2,
  "progress": 500000,
  "status": 0
}
```

属性	说明
currentMusicId	当前背景音乐ID
duration	音频时长, 单位: 毫秒
Status	播放暂停状态
policy	播放策略
progress	播放进度 0~1000000

# remove-bgm-from-list接口

从BGM播放列表移除某个音频

## 请求方式

```
POST http://ip/mwapi/remove-bgm-from-list
```

参数	说明
id	资源ID

```
{  
  "id":203  
}
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

# sort-bgm-list 接口

改变播放列表顺序

## 请求方式

POST http://ip/mwapi/sort-bgm-list

参数	说明
id	背景音乐ID
position	移动后位置 (在新列表中的下标, 0,1,2...)

```
{  
  "id":203,"position":0  
}
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

# switch-bgm接口

切换背景音乐

## 请求方式

POST http://ip/mwapi/switch-bgm

参数	说明
id	背景音乐ID

```
{  
  "id":203  
}
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

# create-scene-with-source 接口

创建含指定源的画面

## 请求方式

POST http://ip/mwapi/get-scenes-list

参数	说明
sourceId	资源id

## 返回数据

JSON 结构如下:

```
{
  "status": 0,
  "message": "success",
  "info": {...}
}
```

### 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 <a href="#">API 状态码</a> 。

### 2. 新建的画面数据 info {...}

```
"info": {
  "audioList": [
    {
      "layerId": 0,
      "muted": 0,
      "volume": 1000000
    }
  ],
  "dividerLine": {
    "color": 0,
    "width": 0,
    "x1": 500000,
    "x2": 500000,
    "y1": 0,
    "y2": 1000000
  },
  "id": 104,
  "layerList": [
    {
      "cropWindow": {
        "height": 1000000,
        "type": 1,
        "width": 1000000,
        "x": 0,
        "y": 0
      },
      "flip": 0,
      "frame": {
        "color": 16777215,
        "width": 0
      },
      "given": 0,
      "givenBg": 0,
      "rotation": 0,
    }
  ]
}
```

```

        "sourceDuration": 0,
        "sourceHeight": 1461,
        "sourceId": 4,
        "sourceType": 1,
        "sourceWidth": 2224,
        "srcWindow": {
            "height": 1000000,
            "width": 856262,
            "x": 71869,
            "y": 0
        },
        "window": {
            "height": 1000000,
            "width": 1000000,
            "x": 0,
            "y": 0
        }
    },
    {
        "cropWindow": {
            "height": 1000000,
            "type": 0,
            "width": 1000000,
            "x": 0,
            "y": 0
        },
        "flip": 0,
        "frame": {
            "color": 0,
            "width": 0
        },
        "givenBg": 0,
        "rotation": 0,
        "sourceHeight": 0,
        "sourceId": 0,
        "sourceType": 0,
        "sourceWidth": 0,
        "srcWindow": {
            "height": 1000000,
            "width": 1000000,
            "x": 0,
            "y": 0
        },
        "window": {
            "height": 0,
            "width": 0,
            "x": 0,
            "y": 0
        }
    }
],
"name": "默认图片",
"template": 3,
"type": 3
}

```

属性	说明
id	场景ID。
name	场景名称。
type	场景类型 4:笔记 其它:场景画面
template	组合模板 0:画中画 1:画面分割 2:单图层画面
dividerLine	分割线属性
layerList	图层列表

## 2. 分割线 dividerLine

属性	说明
color	分割线颜色 uint32 (gbra)

x1	分割线顶点x值
y1	分割线顶点y值
x2	分割线结束点x值
y2	分割线结束点y值

### 3. 场景

属性	说明
sourceId	源ID
sourceType	源类型 1:图片; 2:视频; 3:HDMI 1; 4:HDMI 2; 5:USB-Camera; 6:音频; 7:背景图片; 8:自动摄像头
sourceWidth	源画面宽度
sourceHeight	源画面高度
sourceDuration	源视频长度
mediaProgress	源播放进度, 仅针对VIDEO类型
rotation	旋转角度, 逆时针方向, 范围[0~360]
flip	翻转属性, 0:不翻转 1:左右翻转 2:上下翻转 3:左右翻转+上下翻转
frame	边框属性
window	窗口属性
srcWindow	内容窗口属性
cropWindow	裁剪窗口属性

### 4. 窗口 window

属性	说明
x	x, 相对值 (0~1) * 1000,000,下同
y	y
width	宽度
height	高度
type	非必要参数, 裁剪类型 0:自由比例 1:原始比例 2:4比3 3:16比9 4:1比1

# delete-scene 接口

删除场景

## 请求方式

POST http://ip/mwapi/delete-scene

参数	说明
sceneIds	场景ID,多个以英文“,”隔开
showId	演示稿ID

```
{
  "sceneIds": "100",
  "showId": 7
}
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

### 2. 删除的画面正在被编辑, (此次请求序列中其它非编辑画面成功删除)

```
{
  "info": {
    "editSceneId": 112
  },
  "message": "device busy",
  "status": 11
}
```



# get-device-status 接口

获取设备状态信息

## 请求方式

POST http://ip/mwapi/get-device-status

## 返回数据

JSON 结构如下:

```
{
  "result": 0,
  "message": "success",
  "info": {...}
}
```

### 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 <a href="#">API 状态码</a> 。

### 2. 基础状态信息 info {...}

```
"info": {
  "annotationStatus": {
    "arrowWidth": 4,
    "boardMode": 2,
    "eraserWidth": 34,
    "fillColor": 0,
    "highlighterWidth": 20,
    "inkDuration": 2000,
    "inkStyle": 0,
    "laserpointerPosX": 142089,
    "laserpointerPosY": 675347,
    "lineWidth": 2,
    "lowerFactor": 0,
    "shapeType": 4,
    "strokeColor": 44230911,
    "upperFactor": 500000,
    "useStroke": false
  },
  "ftbEnable": 0,
  "musicStatus": {
    "currentMusicId": 201,
    "duration": 253000,
    "policy": 4,
    "progress": 147919,
    "status": 1
  },
  "recordStatus": {
    "lastRecordError": 0,
    "recording": 1,
    "timeRecording": 18,
    "timeRemaining": 84640
  },
  "sceneStatus": {
    "cropping": 0,
    "croppingLayerId": 0,
    "currentID": 246,
    "currentShowId": 4,
    "editing": 0,

```

```

    "currentShowId": 7,
    "defaultNoteBgId": 1001,
    "isFrozen": 0,
    "lastNoteId": 114,
    "lastSceneId": 246,
    "noteEditing": 0
  },
  "srtStatus": [
    {
      "port": 9110,
      "srtId": 0,
      "stoped": 0
    },
    {
      "port": 9111,
      "srtId": 1,
      "stoped": 0
    },
    {
      "port": 9112,
      "srtId": 2,
      "stoped": 0
    },
    {
      "port": 9113,
      "srtId": 3,
      "stoped": 0
    },
    {
      "port": 9114,
      "srtId": 4,
      "stoped": 0
    }
  ],
  "videoPlayerStatus": [
    {
      "layerId": 0,
      "playStatus": 0,
      "progress": 801324,
      "sourceDuration": 18875,
      "sourceId": 1191
    }
  ]
},

```

属性	说明
annotationStatus	注解配置
ftbEnable	是否启用FTB 0:未启用 1:已启用
musicStatus	BGM播放器状态信息
recordStatus	录制状态信息
sceneStatus	当前画面状态信息
srtStatus	采集画面SRT流服务状态信息
videoPlayerStatus	当前画面Video播放器状态信息

#### BGM播放器状态信息 musicStatus

属性	说明
currentMusicId	当前播放音乐ID
duration	音频文件时长, (单位:毫秒)
policy	播放策略, 1:顺序播放 2:随机播放 3:单曲循环 4:整体循环
progress	播放进度 * 1000,1000
status	播放暂停状态 0:暂停 1:播放

#### 录制状态信息 recordStatus

属性	说明
----	----

lastRecordError	最后一次录制异常信息
recording	录制任务状态, 0:未启用 1:录制中
timeRecording	录制持续时间, 秒
timeRemaining	可用录制时间, 秒

#### 当前画面状态信息 sceneStatus

属性	说明
currentShowId	当前演示稿ID
currentID	当前预览画面ID
lastNoteId	最后一次预览的笔记画面ID
lastSceneId	最后一次预览的场景画面ID
editing	是否处于画面编辑状态, 0:false 1:true
editSceneId	编辑的场景ID
defaultNoteBgId	默认笔记背景资源ID
cropping	是否处于图层裁剪编辑状态, 0:false 1:true
croppingLayerId	裁剪编辑的图层ID
isFrozen	预览画面是否被冻结, 0:false 1:true

#### 当前画面Video播放器状态信息 videoPlayerStatus

属性	说明
layerId	图层ID
playStatus	播放状态, 0:暂停 1:播放
progress	播放进度, 0~1000000
sourceDuration	视频文件持续时长, 毫秒
sourceId	视频文件资源ID

#### SRT推流配置

属性	说明
srtId	srtId 0:预览画面; 1:HDMI 1; 2:HDMI 2; 3:USB Camera
port	端口号
stoped	启停状态 1:停止 0:启动





# get-scenes-list 接口

获取场景列表

## 请求方式

POST http://ip/mwapi/get-scenes-list

## 返回数据

JSON 结构如下:

```
{
  "status": 0,
  "info": {
    "totalCount": 3,
    "pageIndex": 0,
    "audioList": [{...}],
    "sceneList": [{...}]
  }
}
```

### 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 <a href="#">API 状态码</a> 。
totalCount	scene items 总数

### 2. 演示稿音频配置 audioList [{...}]

```
"audioList": [
  {
    "muted": 0,
    "type": 1,
    "volume": 1000000
  },
  {
    "muted": 1,
    "type": 2,
    "volume": 1000000
  },
  {
    "muted": 0,
    "type": 3,
    "volume": 1000000
  },
  {
    "muted": 0,
    "type": 4,
    "volume": 1000000
  },
  {
    "muted": 0,
    "type": 5,
    "volume": 1000000
  },
  {
    "muted": 0,
    "type": 6,
    "volume": 1000000
  }
],
```

属性	说明
type	音频类型 1:output 2:monitor 3:mic 4:bgm 5:uac in 6:line in
muted	静音 0:NO 1:YES
volume	音量 0~1000000

### 3. 当前分页场景列表 sceneList [{...}]

```

"sceneList":[
  {
    "audiolist": [
      {
        "layerId": 0,
        "muted": 0,
        "type": 256,
        "volume": 1000000
      }
    ],
    "dividerLine": {
      "color": 0,
      "width": 0,
      "x1": 500000,
      "x2": 500000,
      "y1": 0,
      "y2": 1000000
    },
    "id": 100,
    "layerList": [
      {
        "cropWindow": {
          "height": 1000000,
          "type": 0,
          "width": 1000000,
          "x": 0,
          "y": 0
        },
        "flip": 0,
        "frame": {
          "color": 0,
          "width": 0
        },
        "givenBg": 0,
        "mediaProgress": 0,
        "rotation": 360,
        "sourceDuration": 0,
        "sourceHeight": 1080,
        "sourceId": 1,
        "sourceThumbHeight": 144,
        "sourceThumbWidth": 256,
        "sourceType": 3,
        "sourceWidth": 1920,
        "srcWindow": {
          "height": 1000000,
          "width": 1000000,
          "x": 0,
          "y": 0
        },
        "window": {
          "height": 1000000,
          "width": 1000000,
          "x": 0,
          "y": 0
        }
      }
    ],
    {
      "cropWindow": {
        "height": 1000000,
        "type": 0,
        "width": 1000000,
        "x": 0,
        "y": 0
      }
    }
  }
]

```

```

    },
    "flip": 0,
    "frame": {
        "color": 0,
        "width": 0
    },
    "givenBg": 0,
    "rotation": 0,
    "sourceHeight": 0,
    "sourceId": 0,
    "sourceThumbHeight": 0,
    "sourceThumbWidth": 0,
    "sourceType": 0,
    "sourceWidth": 0,
    "srcWindow": {
        "height": 1000000,
        "width": 1000000,
        "x": 0,
        "y": 0
    },
    "window": {
        "height": 0,
        "width": 0,
        "x": 0,
        "y": 0
    }
}
],
"name": "",
"template": 2,
"type": 3
},
{
    "audiolist": [],
    "dividerLine": {
        "color": 0,
        "width": 0,
        "x1": 500000,
        "x2": 500000,
        "y1": 0,
        "y2": 1000000
    },
    "id": 101,
    "layerList": [
        {
            "cropWindow": {
                "height": 1000000,
                "type": 0,
                "width": 1000000,
                "x": 0,
                "y": 0
            },
            "flip": 0,
            "frame": {
                "color": 0,
                "width": 0
            },
            "givenBg": 0,
            "rotation": 0,
            "sourceHeight": 1080,
            "sourceId": 1001,
            "sourceThumbHeight": 0,
            "sourceThumbWidth": 0,
            "sourceType": 7,
            "sourceWidth": 1920,
            "srcWindow": {
                "height": 1000000,
                "width": 1000000,
                "x": 0,
                "y": 0
            },
            "thumbHeight": 144,
            "thumbWidth": 256,

```



```

        "window": {
            "height": 1000000,
            "width": 1000000,
            "x": 0,
            "y": 0
        }
    },
    {
        "cropWindow": {
            "height": 1000000,
            "type": 0,
            "width": 1000000,
            "x": 0,
            "y": 0
        },
        "flip": 0,
        "frame": {
            "color": 0,
            "width": 0
        },
        "givenBg": 0,
        "rotation": 0,
        "sourceDuration": 0,
        "sourceHeight": 1080,
        "sourceId": 2039,
        "sourceThumbHeight": 0,
        "sourceThumbWidth": 0,
        "sourceType": 1,
        "sourceWidth": 1920,
        "srcWindow": {
            "height": 1000000,
            "width": 1000000,
            "x": 0,
            "y": 0
        },
        "window": {
            "height": 1000000,
            "width": 1000000,
            "x": 0,
            "y": 0
        }
    }
],
"name": "note",
"template": 0,
"type": 4
}
]

```

### 1. 场景

属性	说明
id	场景ID。
name	场景名称。
type	场景类型 4:笔记 其它:场景画面（不再细分类型）
template	组合模板 0:画中画 1:画面分割 2:单图层画面
dividerLine	分割线属性
layerList	图层列表

### 2. 场景内音频配置

属性	说明
type	音频类型 0x0100: HDMI 1; 0x0200: video; 0x0300: HDMI 2; 0x0400: USB Camera; 0x0500:picture
muted	静音 0:NO 1:YES
volume	音量 0~1000000

### 3. 分割线 dividerLine

属性	说明
color	分割线颜色 uint32 (gbra)
x1	分割线顶点x值
y1	分割线顶点y值
x2	分割线结束点x值
y2	分割线结束点y值

#### 4. 图层 layer

属性	说明
sourceId	源ID。可通过HTTP访问 ( <a href="http://[设备IP]/source/(sourceId)">http://[设备IP]/source/(sourceId)</a> )
sourceType	源类型 1: 图片; 2: 视频; 3: HDMI 1; 4: HDMI 2; 5: USB-Camera; 6: 音频; 7: 背景图片; 8: 自动摄像头
sourceWidth	源画面宽度
sourceHeight	源画面高度
sourceDuration	源视频长度
mediaProgress	源播放进度, 仅针对VIDEO类型
rotation	旋转角度, 逆时针方向, 范围[0~360]
flip	翻转属性, 0:不翻转 1:左右翻转 2:上下翻转 3:左右翻转+上下翻转
frame	边框属性
window	窗口属性
srcWindow	内容窗口属性
cropWindow	裁剪窗口属性

#### 5. 窗口 window

属性	说明
x	x, 相对值 (0~1) * 1000,000,下同
y	y
width	宽度
height	高度
type	非必要参数, 裁剪类型 0:自由比例 1:原始比例 2:4比3 3:16比9 4:1比1

# get-source-thumbnail 接口

获取资源缩略图

## 请求方式

POST http://ip/mwapi/get-source-thumbnail

参数	说明
sourceId	资源ID
mediaProgress	进度*1000,000

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

### 2. thumbnail内容

```
"thumbnail": {
  "encoding": "base64",
  "data": "..."}
}
```

属性	说明
encoding	加密方式。
data	加密后数据

### HTTP访问缩图

[http://\[IP\]/source/\[sourceId\]](http://[IP]/source/[sourceId])

# get-video-player-status 接口

获取当前演示场景视频播放状态信息

## 请求方式

POST http://ip/mwapi/get-video-player-status

## 返回数据

JSON 结构如下:

```
{
  "status":0,
  "info":[
    {
      "layerId":1,
      "playStatus":1,
      "progress":16664,
      "sourceDuration":120000,
      "sourceId":101
    }
  ]
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
layerId	图层ID, 即图层处于layerList的下标
sourceId	资源ID
playStatus	播放状态 0:pause 1:playing
progress	播放进度, 0~1000000
sourceDuration	视频时长, 单位毫秒

# rename-scene 接口

重命名画面

## 请求方式

POST http://ip/mwapi/rename-scene

参数	说明
sceneId	编辑的场景ID
name	新的名称
showId	演示稿ID

```
{  
  "sceneId":100,  
  "name":"默认场景12we3",  
  "showId":7  
}
```

## 返回数据

JSON 结构如下:

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

### 1. 返回状态

```
{  
  "message": "Scene is not in current presentation",  
  "status": 48  
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。

# seek-video-player接口

设置当前场景视频播放进度

## 请求方式

POST http://ip/mwapi/seek-video-player

参数	说明
layerId	图层ID
progress	播放进度, 0~1000000

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

# set-switch-action 接口

触发切换按钮

## 请求方式

```
POST http://ip/mwapi/set-switch-action
```

```
{  
  "action": 0,  
}
```

参数	说明
action	0~4,对应设备上的按键

## 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 <a href="#">API 状态码</a> 。
action	按键事件(0-4)。

# set-video-player-status接口

设置当前场景视频播放状态

## 请求方式

```
POST http://ip/mwapi/set-video-player-status
```

参数	说明
layerId	图层ID
playStatus	播放状态 0:pause 1:playing

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。



# switch-scene 接口

切换画面

## 请求方式

```
POST http://ip/mwapi/switch-scene
```

参数	说明
sceneld	场景ID

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

### 2. 设备正处于场景编辑状态, 不允许切换场景

```
{  
  "message": "device busy",  
  "status": 11  
}
```

## upload-source-file 接口

上传源文件，上传文件为图片、视频或音乐。

### 请求方式

```
POST http://ip/mwapi/upload-source-file
```

### 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 <a href="#">API 状态码</a> 。

# get-app-settings 接口

获取 APP 相关设置。

## 请求方式

POST http://ip/mwapi/get-app-settings

## 返回数据

```
{
  "status": 0,
  "need-password": 0,
  "have-password": 0,
  "ssdp-port": 1900,
  "control-port": 9000,
  "stream-port": 9110,
  "ftp-port": 21,
  "resolution": "1280x720",
  "duration": 333333,
  "video-bitrate": 2048
}
```

属性	说明
status	返回状态。0: 上传成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
need-password	是否需要验证密码 0:无 1:有
have-password	密码是否为空 0:空 1:非空
resolution	分辨率
duration	帧率
video-bitrate	编码位率
stream-port	srt推流端口

# get-encoder-params 接口

获取编码参数集合。

## 请求方式

POST http://ip/mwapi/get-encoder-params

## 返回数据

```
{
  "status": 0,
  "resolutions": [{...}],
  "durations": [{...}],
  "video-bitrates": [{...}],
}
```

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

### 1.分辨率 resolutions

```
[
  {
    "name": "1080P",
    "value": "1920x1080"
  },
  {
    "name": "720P",
    "value": "1080x720"
  }
]
```

属性	说明
name	显示名称
value	取值: 宽x高

### 2.帧间隔 durations

```
[
  {
    "name": "60 FPS",
    "value": "166667"
  },
  {
    "name": "59.94 FPS",
    "value": "166833"
  },
  {
    "name": "50 FPS",
    "value": "200000"
  },
  {
    "name": "30 FPS",
    "value": "333333"
  },
  {
    "name": "29.97 FPS",
    "value": "333667"
  },
  {

```

```

    "name": "25 FPS",
    "value": "400000"
  },
  {
    "name": "24 FPS",
    "value": "416667"
  },
  {
    "name": "15 FPS",
    "value": "666667"
  },
  {
    "name": "10 FPS",
    "value": "1000000"
  },
  {
    "name": "5 FPS",
    "value": "2000000"
  }
]

```

属性	说明
name	显示名称
value	取值, 帧间隔

### 3. 视频编码位率 video-bitrates

```

[
  {
    "name": "8 Mbps",
    "value": "8192"
  },
  {
    "name": "6 Mbps",
    "value": "6144"
  },
  {
    "name": "4 Mbps",
    "value": "4096"
  },
  {
    "name": "2 Mbps",
    "value": "2048"
  },
  {
    "name": "1 Mbps",
    "value": "1024"
  },
  {
    "name": "768 Kbps",
    "value": "768"
  },
  {
    "name": "512 Kbps",
    "value": "512"
  },
  {
    "name": "256 Kbps",
    "value": "256"
  }
]

```

属性	说明
name	显示名称
value	取值, 视频编码位率



## set-app-password 接口

设置 APP 端验证密码。

### 请求方式

POST http://ip/mwapi/set-app-password

参数	说明
method	方法名称: set-app-password
need-password	是否需要密码 0:无需密码 1:需要密码
new-password	新密码, 设置需要密码时必填

```
{
  "new-password": "8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92",
  "need-password": 1
}
```

### 返回数据

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

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

## set-encoder-format 接口

修改编码参数。

### 请求方式

POST http://ip/mwapi/set-encoder-format

参数	说明
method	方法名称: set-encoder-format
resolution	分辨率
duration	帧率
video-bitrate	视频编码位率

```
{
  "resolution": "1920x1080",
  "duration": 333333,
  "video-bitrate": 2048
}
```

### 返回数据

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

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



## set-network-port 接口

修改网络协议端口。

### 请求方式

POST http://ip/mwapi/set-network-port

参数	说明
control-port	控制命令网络协议端口
stream-port	srt推流协议网络端口

```
{
  "control-port":9000,
  "stream-port":9110
}
```

### 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 <a href="#">API 状态码</a> 。

## /system/device-info 接口

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

### 请求方式

POST /api/system/device-info

### 返回数据

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

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

## /system/info 接口

获取cpu, 内存信息等信息。

### 请求方式

POST /api/system/info

### 返回数据

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

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

## /system/set-device-name 接口

设置设备名称。

### 请求方式

POST /api/system/set-device-name

参数	说明
name	设备名称。

### 返回数据

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

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

## /system/timezone-set 接口

时区设置。

### 请求方式

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

参数	说明
zonename	时区名称。

### 返回数据

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

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

## /system/set-date-time 接口

NTP 功能设置。

### 请求方式

```
POST /api/system/set-date-time
```

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

### 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 <a href="#">API 状态码</a> 。

## /user/login 接口

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

### 请求方式

POST /api/user/login

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

### 返回数据

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

属性	说明
status	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 <a href="#">API 状态码</a> 。

### 接口示例

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

## /user/logout 接口

退出登录，返回到登录界面。

### 请求方式

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

### 返回数据

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

属性	说明
status	返回状态。0：退出登录成功，返回其它值请参考 <a href="#">API 状态码</a> 。



## /user/get-all 接口

获取系统用户列表信息，仅管理员有权限。

### 请求方式

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

### 返回数据

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

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

## /user/add 接口

添加用户，仅管理员有权限。

### 请求方式

POST /api/user/add

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

### 返回数据

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

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

## /user/del 接口

删除用户，仅管理员有权限。

### 请求方式

POST /api/user/del

参数	说明
username	用户登录名。

### 返回数据

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

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

## /user/ch-password 接口

用户修改自己的登录密码，修改时必须输入原密码。

### 请求方式

POST /api/user/ch-password

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

### 返回数据

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

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

## /user/set-password 接口

重置用户密码，无需输入原密码，仅管理员有权限。

### 请求方式

POST /api/user/set-password

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

### 返回数据

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

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

## /network/if-info 接口

网卡信息。

### 请求方式

POST /api/network/if-info

### 返回数据

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

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

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

## /network/if-set 接口

配置网卡。

### 请求方式

```
POST /api/network/if-set
```

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

### 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 <a href="#">API 状态码</a> 。



## /network/if-route 接口

获取默认路由。

### 请求方式

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

### 返回数据

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

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

## /network/get-dns 接口

获取DNS。

### 请求方式

POST /api/network/get-dns

### 返回数据

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

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

## /network/set-dns 接口

设置DNS。

### 请求方式

POST /api/network/set-dns

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

### 返回数据

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

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 <a href="#">API 状态码</a> 。

## /network/usb-config 接口

配置USB网卡。

### 请求方式

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

参数	说明
iface	网卡名称。
ipaddr	IP 地址。

### 返回数据

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

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 <a href="#">API 状态码</a> 。

## /network/wireless-mode 接口

WIFI 工作模式切换。

### 请求方式

```
POST /api/network/wireless-mode
```

参数	说明
iface	网卡名称。
mode	WIFI 工作模式 0: STA 模式 1: AP 模式。

### 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 <a href="#">API 状态码</a> 。

## /network/wireless-state 接口

AP 工作状态。

### 请求方式

```
POST /api/network/wireless-state
```

参数	说明
iface	网卡名称。

### 返回数据

```
{
  "enable": true,
  "mode": 1,
  "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 <a href="#">API 状态码</a> 。
mode	WIFI 工作模式 0: STA 模式 1: AP 模式。
enable	网卡服务是否开启。

## /network/ap-passwd 接口

设置AP密码。

### 请求方式

POST /api/network/ap-passwd

参数	说明
iface	网卡名称。
password	AP密码。

### 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 <a href="#">API 状态码</a> 。

## /network/ap-config 接口

设置AP。

### 请求方式

POST /api/network/ap-config

参数	说明
iface	网卡名称。
ssid	AP 名称。
hw_mode	wifi 工作频率 a: 5GHz g: 2.4GHz。
channel	AP 默认通道。
country-code	参考ISO/IEC 3166-1。
ipaddr	IP地址。
dhcp-ip-start	DHCP 分配的IP起始地址。
dhcp-ip-end	DHCP 分配的IP结束地址。
enable-surfing	使能冲浪上网。

### 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 <a href="#">API 状态码</a> 。



## /network/ap-get-config 接口

获取AP 信息。

### 请求方式

POST /api/network/ap-get-config

参数	说明
iface	网卡名称。

### 返回数据

```
{
  "ssid": "USB-Fusion_3002",
  "hw_mode": "g",
  "channel": 1,
  "country-code": "CN",
  "support-freq": [
    {
      "freq": "2.4G",
      "hw-mode": "g"
    },
    {
      "freq": "5G",
      "hw-mode": "a"
    }
  ],
  "support-channels": {
    "channels": [
      {
        "code": [
          "US",
          "CA"
        ],
        "2g-chans": [
          1,
          2,
          3,
          4,
          5,
          6,
          7,
          8,
          9,
          10,
          11
        ],
        "5g-chans": [
          36,
          40,
          44,
          48,
          149,
          153,
          157,
          161,
          165
        ]
      }
    ],
    {
      "code": [
        "AL",
        "AM",
        "AT",
        "AZ",
```

```

    "BA",
    "BE",
    "BG",
    "BY",
    "CH",
    "CY",
    "CZ",
    "DE",
    "DK",
    "EE",
    "EL",
    "ES",
    "FI",
    "FR",
    "GE",
    "HR",
    "HU",
    "IE",
    "IS",
    "IT",
    "LI",
    "LT",
    "LU",
    "LV",
    "MD",
    "ME",
    "MK",
    "MT",
    "NL",
    "NO",
    "PL",
    "PT",
    "RO",
    "RS",
    "RU",
    "SE",
    "SI",
    "SK",
    "TR",
    "UA",
    "UK"
  ],
  "2g-chans": [
    1,
    2,
    3,
    4,
    5,
    6,
    7,
    8,
    9,
    10,
    11,
    12,
    13
  ],
  "5g-chans": [
    36,
    40,
    44,
    48,
    149,
    153,
    157,
    161,
    165
  ]
},
{
  "code": [
    "JP"
  ],

```

```

    "2g-chans": [
      1,
      2,
      3,
      4,
      5,
      6,
      7,
      8,
      9,
      10,
      11,
      12,
      13
    ],
    "5g-chans": [
      36,
      40,
      44,
      48
    ]
  ],
  {
    "code": [
      "CN"
    ],
    "2g-chans": [
      1,
      2,
      3,
      4,
      5,
      6,
      7,
      8,
      9,
      10,
      11,
      12,
      13
    ],
    "5g-chans": [
      36,
      40,
      44,
      48,
      149,
      153,
      157,
      161,
      165
    ]
  }
],
"global-2g-chans": [
  1,
  2,
  3,
  4,
  5,
  6,
  7,
  8,
  9,
  10,
  11,
  12,
  13
],
"global-5g-chans": [
  36,
  40,
  44,

```

```

        48,
        149,
        153,
        157,
        161,
        165
    ]
},
"ipaddr": "192.168.67.1",
"dhcp-ip-start": "192.168.67.2",
"dhcp-ip-end": "192.168.67.254",
"first-use": false,
"ap-to-sta-reboot-effect": true,
"sta-to-ap-reboot-effect": false,
"enable-surfing": true,
"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 <a href="#">API 状态码</a> 。
ssid	AP 名称。
hw_mode	wifi 工作频率 a: 5GHz g: 2.4GHz。
channel	AP 默认通道。
country-code	参考ISO/IEC 3166-1。
support-freq	支持的频率。
ipaddr	IP地址。
dhcp-ip-start	DHCP 分配的IP起始地址。
dhcp-ip-end	DHCP 分配的IP结束地址。
enable-surfing	使能冲浪上网。
first-use	首次切换到AP模式。
ap-to-sta-reboot-effect	AP切换STA是否需要reboot。
sta-to-ap-reboot-effect	STA切换AP是否需要reboot。

## /network/wifi-scan 接口

WIFI 扫描周围可连接的热点。

### 请求方式

```
POST /api/network/wifi-scan
```

参数	说明
iface	网卡名称。

### 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 <a href="#">API 状态码</a> 。

## /network/wifi-scan-results 接口

获取wifi-scan 的结果。

### 请求方式

POST /api/network/wifi-scan-results

参数	说明
iface	网卡名称。

### 返回数据

```
{
  "connect": {
    "ssid": "magewell",
    "state": 2
  },
  "user-items": [
    {
      "ssid": "magewell",
      "bssid": "94:a6:7e:61:75:c6",
      "frequency": 5220,
      "signal": -48,
      "encryption": "WPA2PSK",
      "is-auto": false,
      "is-used": true
    }
  ],
  "scan-items": [
    {
      "ssid": "TP-LINK_8EF4",
      "bssid": "18:f2:2c:1e:8e:f4",
      "frequency": 2412,
      "signal": -39,
      "encryption": "WPA2PSK",
      "is-auto": false,
      "is-used": false
    },
    {
      "ssid": "magewell_005_5GHz",
      "bssid": "14:91:82:2b:1f:6d",
      "frequency": 5765,
      "signal": -40,
      "encryption": "WPAPSK",
      "is-auto": false,
      "is-used": false
    },
    {
      "ssid": "TP-LINK_5G_8EF4",
      "bssid": "18:f2:2c:1e:8e:f6",
      "frequency": 5200,
      "signal": -42,
      "encryption": "WPA2PSK",
      "is-auto": false,
      "is-used": false
    },
    {
      "ssid": "magewell_001",
      "bssid": "f0:b4:29:54:8e:b5",
      "frequency": 2412,
      "signal": -43,
      "encryption": "WPAPSK",
      "is-auto": false,
      "is-used": false
    }
  ]
}
```

```

    ],
    "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 <a href="#">API 状态码</a> 。
connect.ssid	连接ssid。
connect.state	AP连接状态 0: 空闲, 无连接, 1: 连接中, 请稍后, 2: 连接成功, 3: 密码错误, 4: 连接失败, 5:连接超时。
user-items[i].ssid	AP 名称。
user-items[i].bssid	AP bssid。
user-items[i].frequency	AP 频率。
user-items[i].encryption	AP加密方式 NONE: 不加密, WEP : WEP 加密方式, WPAPSK: WPAPSK 加密方式, WPA2PSK: WPA2PSK 加密方式。
user-items[i].is-auto	是否允许自动连接。
user-items[i].is-used	是否连接成功。
scan-items[i].ssid	AP 名称。
scan-items[i].bssid	AP bssid。
scan-items[i].frequency	AP 频率。
scan-items[i].encryption	AP加密方式 NONE: 不加密, WEP : WEP 加密方式, WPAPSK: WPAPSK 加密方式, WPA2PSK: WPA2PSK 加密方式。
scan-items[i].is-auto	是否允许自动连接。
scan-items[i].is-used	是否连接成功。

## /network/wifi-connect 接口

连接WIFI。

### 请求方式

```
POST /api/network/wifi-connect
```

参数	说明
iface	网卡名称。
ssid	AP 名称。
password	AP 连接密码, 若填写空字符, 表示连接未加密的AP 或者user-items有其他已连过的AP。
is-auto	是否自动连接。

### 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 <a href="#">API 状态码</a> 。



## /network/wifi-connect-auto 接口

设置网络自动连接。

### 请求方式

POST /api/network/wifi-connect-auto

参数	说明
iface	网卡名称。
ssid	AP 名称。
is-auto	是否自动连接。

### 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 <a href="#">API 状态码</a> 。

## /network/wifi-connect-state 接口

WIFI连接AP状态。

### 请求方式

POST /api/network/wifi-connect-state

参数	说明
iface	网卡名称。

### 返回数据

```
{
  "ssid": "magewell",
  "state": 2,
  "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 <a href="#">API 状态码</a> 。
ssid	连接的AP SSID。
state	AP连接状态 0: 空闲, 无连接, 1: 连接中, 请稍后, 2: 连接成功, 3: 密码错误, 4: 连接失败, 5:连接超时。

## /network/wifi-disconnect 接口

断开连接AP。

### 请求方式

POST /api/network/wifi-disconnect

参数	说明
iface	网卡名称。
ssid	AP 名称。

### 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 <a href="#">API 状态码</a> 。

## /network/wifi-forget 接口

移除网络配置。

### 请求方式

```
POST /api/network/wifi-forget
```

参数	说明
iface	网卡名称。
ssid	AP 名称。

### 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 <a href="#">API 状态码</a> 。

## upgrade/online-check 接口

启动在线升级检查。

### 请求方式

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

### 返回数据

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

属性	说明
status	返回状态。0: 开始更新, 返回其它值请参考 <a href="#">API 状态码</a> 。

# upgrade/online-check-result 接口

获取在线检查结果。

## 请求方式

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

## 返回数据

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

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

## /upgrade/upload-fw 接口

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

### 请求方式

POST /upgrade/upload-fw

### 返回数据

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

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

## /upgrade/update 接口

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

### 请求方式

POST /api/upgrade/update

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

### 返回数据

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

属性	说明
status	返回状态。0: 开始更新, 返回其它值请参考 <a href="#">API 状态码</a> 。



## /upgrade/state 接口

获取当前固件版本信息和升级状态，仅管理员有权限。

### 请求方式

POST /api/upgrade/state

### 返回数据

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

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

## /upgrade/clear 接口

清除升级状态。

### 请求方式

```
POST /upgrade/clear
```

### 返回数据

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

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。

## /log/clear 接口

清除全部系统日志，仅管理员有权限。

### 请求方式

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

### 返回数据

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

# /log/filter 接口

过滤日志。

## 请求方式

POST /api/log/filter

参数	说明
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。
key	过滤关键字, 可以为空字符串。

## 返回数据

```
{
  "status": 0,
  "logs": [
    {
      "time": "2021/12/20 13:18:27.210",
      "type": "info",
      "message": "User 'Admin' (192.168.66.2) logged in"
    },
    {
      "time": "2021/12/20 11:07:29.548",
      "type": "info",
      "message": "Network changed : wlan0(10.20.0.68) connected"
    },
    ...
  ]
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
logs[i].time	时间。
logs[i].type	输出级别 info, warn, error。
logs[i].context	日志内容。

## /log/export 接口

导出设备当前的系统日志，导出文件为 html 格式，仅管理员有权限。

### 请求方式

POST /api//log/export

参数	说明
filename	导出的文件名称

### 请求结果

直接下载 html 格式日志文件到本地。

# MSG\_WEB\_SORKET\_EVENT\_CHANGE\_INFO 接口

通过WebSocket建立连接，监听设备状态变更信息。

## 请求方式

```
ws://xx.xx.xx.xx/ws_up_state
```

## 返回数据

```
{
  "event":4001,
  "info":{
    "statusType":2003
  }
}
```

属性	说明
event	消息事件类型
info	消息对应信息实体

## event类型

属性	Value	说明
MSG_WEB_SORKET_EVENT_START	4000	开始标记，无意义
MSG_WEB_SORKET_EVENT_CHANGE_INFO	4001	设备信息变更事件
MSG_WEB_SORKET_EVENT_END	4002	结束标记，无意义

## statusType类型

Value	说明
2000	演示稿列表变更
2001	演示稿切换
2002	场景列表变更
2003	当前预览画面切换
2004	BGM列表变更
2005	BGM播放状态变更
2006	资源列表变更
2007	相册列表变更
2008	音频设置变更
2009	FTB模式启用状态变更，可使用 /mwapi/get-device-status 获取数据
2010	录制状态变更
2011	场景编辑模式变更,如: {"event":4001,"info":{"editScenelId":104,"editing":1,"statusType":2011}}
2012	资源变更（采集源信号拔插），如: {"event":4001,"info":{"sourceId":1,"statusType":2012}}
2013	开机画面列表变更
2014	当前选中开机画面变更
2015	设备通用设置变更

# get-album-files-list 接口

获取服务器相册（录制或截屏）文件列表

## 请求方式

POST http://ip/mwapi/get-album-files-list

参数	说明
type	资源类型。0: all 1: 截屏图片 2: 录制视频

## 返回数据

JSON 结构如下:

```
{
  "info": {
    "items": [
      {
        "createTime": 1640055676,
        "duration": 0,
        "filePath": "album/2a83adf5-3e0d-4ad0-9af5-81d8fc0fea18.jpeg",
        "id": 2,
        "md5": "md5",
        "name": "2021.12.21 11:01AM",
        "size": 502784,
        "thumbPath": "album/2a83adf5-3e0d-4ad0-9af5-81d8fc0fea18.thumb.jpeg",
        "type": 1
      },
      {
        "createTime": 1640055673,
        "duration": 1849,
        "filePath": "album/16241d75-12ea-4344-bf7f-bf683e9b850e.mp4",
        "id": 1,
        "md5": "md5",
        "name": "2021.12.21 11:01AM",
        "size": 1487872,
        "thumbPath": "album/16241d75-12ea-4344-bf7f-bf683e9b850e.thumb.jpeg",
        "type": 2
      }
    ],
    "pageIndex": 0,
    "totalCount": 2
  },
  "message": "success",
  "result": 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 <a href="#">API 状态码</a> 。
id	文件ID
type	文件类型。1: 截屏图片 2: 录制视频
name	资源名称
md5	文件md5值
filePath	文件相对路径, 可通过http访问 ( <a href="#">http://设备IP/相对路径</a> )
thumbPath	缩略图相对路径, 可通过http访问 ( <a href="#">http://设备IP/相对路径</a> )
createTime	文件创建时间 相对1970-01-01时间戳
duration	文件持续时间。仅视频有效, 单位毫秒
size	文件大小, 单位Byte





## remove-album-file 接口

删除相册文件

### 请求方式

```
POST http://ip/mwapi/remove-album-file
```

参数	说明
ids	文件ID,以英文“,”分割

### 返回数据

JSON 结构如下:

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

#### 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 <a href="#">API 状态码</a> 。

## get-volume-config 接口

获取音量信息。

### 请求方式

POST http://ip/mwapi/get-volume-config

### 返回数据

```
{
  "status": 0,
  "output": [
    {
      "id": 1,
      "db": 0,
      "mute": false
    }
  ],
  "input": [
    {
      "id": 11,
      "db": 0,
      "mute": false
    },
    {
      "id": 14,
      "db": 0,
      "mute": false
    },
    {
      "id": 17,
      "db": 0,
      "mute": false
    },
    {
      "id": 18,
      "db": 0,
      "mute": false
    },
    {
      "id": 12,
      "db": 0,
      "mute": false
    },
    {
      "id": 13,
      "db": 0,
      "mute": false
    },
    {
      "id": 19,
      "db": 0,
      "mute": false
    }
  ]
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
id	1: Output; 11: Mic; 12: HDMI 1; 13: HDMI 2; 14: BGM; 15: Layer 1中的视频; 16: Layer 2中的视频; 17: USB-C; 18: Line in; 19: USB Camera
output	输出音量列表。
input	输入音量列表。



## set-volume-config 接口

设置音量。

### 请求方式

```
POST http://ip/mwapi/set-volume-config
```

参数	说明
id	设置目标
db	音量电平值, (-100~0) db
mute	设置静音

### 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 <a href="#">API 状态码</a> 。

## get-audio-card-list 接口

获取声卡设备列表。

### 请求方式

POST http://ip/mwapi/get-audio-card-list

### 返回数据

```
{
  "status": 0,
  "cards": [
    {
      "card": 0,
      "name": "USB Fusion Audio",
      "system": false
    },
    {
      "card": 3,
      "name": "USB Capture HDMI+",
      "system": false
    }
  ]
}
```

属性	说明
cards	声卡身份唯一标识。
name	声卡设备名称。
system	系统内部的声卡，还是外部接入设备创建的声卡。True: 内部; False: 外部

## get-audio-card-mixer 接口

获取声卡属性。

### 请求方式

POST http://ip/mwapi/get-audio-card-mixer

```
{
  "card": 0,
}
```

参数	说明
card	声卡身份唯一标识

### 返回数据

```
{
  "status": 0,
  "mixer": [
    {
      "name": "Headphone",
      "id": 0,
      "val": 100,
      "flag_c": false,
      "flag_p": true
    },
    {
      "name": "Line Gain",
      "id": 5,
      "val": 100,
      "flag_c": false,
      "flag_p": true
    },
    {
      "name": "Mic Gain",
      "id": 7,
      "val": 100,
      "flag_c": false,
      "flag_p": true
    },
    {
      "name": "Mic",
      "id": 14,
      "val": 100,
      "flag_c": true,
      "flag_p": false
    },
    {
      "name": "Line",
      "id": 15,
      "val": 100,
      "flag_c": true,
      "flag_p": false
    }
  ]
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
mixer	声卡属性列表。

属性	说明
----	----

name	属性名称。
id	属性身份唯一标识。
val	属性值, 0~100
flag_c	是否为采集音量。True: 是 False: 否
flag_p	是否为播放音量。True: 是 False: 否

## get-mic-gain接口

获取麦克风增益是否启用。

### 请求方式

```
POST http://ip/mwapi/get-mic-gain
```

### 返回数据

```
{  
  "status": 0,  
  "gain": 1  
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
gain	0: 未启用 1: 已启用



## get-mic-monitor 接口

获取监听麦克风启用状态。

### 请求方式

```
POST http://ip/mwapi/get-mic-monitor
```

### 返回数据

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

属性	说明
monitor	是否监听麦克风。 true: 监听, false: 不监听

## get-usb-audio-list 接口

获取USB音频设备列表。

### 请求方式

```
POST http://ip/mwapi/get-usb-audio-list
```

```
{
  "target": 0,
}
```

参数	说明
target	0: 选择 Webcam 音频, 1: 选择 USB 麦克风音频, 2: 选择 USB 输出音频。

### 返回数据

```
{
  "status": 0,
  "selected": false,
  "cur-dev-path": "",
  "devices": [
    {
      "dev-path": "aHc6Mywx",
      "dev-name": "HDMI [USB Capture HDMI+] #1"
    },
    {
      "dev-path": "aHc6Mywy",
      "dev-name": "HDMI [USB Capture HDMI+] #2"
    }
  ]
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
selected	是否已选择设备
cur-dev-path	当前设备
devices	设备列表

## set-audio-card-mixer 接口

设置声卡属性。

### 请求方式

```
POST http://ip/mwapi/set-audio-card-mixer
```

参数	说明
card	声卡身份唯一标识
id	声卡属性身份唯一标识
vol	属性值, 0~100

### 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 <a href="#">API 状态码</a> 。

## set-mic-gain接口

设置麦克风增益启用状态。

### 请求方式

```
POST http://ip/mwapi/set-mic-gain
```

```
{  
  "gain": 0,  
}
```

参数	说明
gain	0: 未启用 1: 已启用

### 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 <a href="#">API 状态码</a> 。

## set-mic-monitor 接口

设置麦克风监听状态。

### 请求方式

```
POST http://ip/mwapi/set-mic-monitor
```

```
{  
  "monitor": true  
}
```

参数	说明
monitor	是否监听麦克风。True: 监听 False: 不监听

### 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 <a href="#">API 状态码</a> 。

## set-usb-audio 接口

选择可应用到整个演示的全局 USB 麦克风和 USB 音频输出设备。

### 请求方式

```
POST http://ip/mwapi/set-usb-audio
```

参数	说明
target	0:web camera 音频设备 1:USB麦克风音频设备 2:USB音频输出设备
dev-path	设备唯一标识, 可使用 <a href="#">get-usb-audio-list</a> 接口获取设备列表

### 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 <a href="#">API 状态码</a> 。

## add-launch-scene 接口

新建开机画面场景。

### 请求方式

POST http://ip/mwapi/add-launch-scene

参数	说明
sourceId	资源ID

```
{
  "sourceId":4
}
```

### 返回数据

```
{
  "info": {
    "audioList": [
      {
        "layerId": 0,
        "muted": 0,
        "volume": 1000000
      }
    ],
    "dividerLine": {
      "color": 0,
      "width": 0,
      "x1": 500000,
      "x2": 500000,
      "y1": 0,
      "y2": 1000000
    },
    "id": 25,
    "layerList": [
      {
        "cropWindow": {
          "height": 1000000,
          "type": 1,
          "width": 1000000,
          "x": 0,
          "y": 0
        },
        "flip": 0,
        "frame": {
          "color": 16777215,
          "width": 0
        },
        "given": 0,
        "givenBg": 0,
        "rotation": 360,
        "sourceDuration": 0,
        "sourceHeight": 1461,
        "sourceId": 4,
        "sourceThumbHeight": 256,
        "sourceThumbWidth": 168,
        "sourceType": 1,
        "sourceWidth": 2224,
        "srcWindow": {
          "height": 1000000,
          "width": 856262,
          "x": 71869,
          "y": 0
        }
      }
    ]
  }
}
```

```

        "videoPolicy": 1,
        "window": {
            "height": 1000000,
            "width": 1000000,
            "x": 0,
            "y": 0
        }
    },
    {
        "cropWindow": {
            "height": 1000000,
            "type": 0,
            "width": 1000000,
            "x": 0,
            "y": 0
        },
        "flip": 0,
        "frame": {
            "color": 0,
            "width": 0
        },
        "givenBg": 0,
        "rotation": 0,
        "sourceHeight": 0,
        "sourceId": 0,
        "sourceThumbHeight": 0,
        "sourceThumbWidth": 0,
        "sourceType": 0,
        "sourceWidth": 0,
        "srcWindow": {
            "height": 1000000,
            "width": 1000000,
            "x": 0,
            "y": 0
        },
        "window": {
            "height": 0,
            "width": 0,
            "x": 0,
            "y": 0
        }
    }
],
"name": "default",
"template": 2,
"type": 3
},
"message": "success",
"status": 0
}

```

属性	说明
status	返回状态。0: 上传成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
info	场景实体



## get-server-launch-scenes 接口

获取开机画面场景列表

### 请求方式

```
POST http://ip/mwapi/get-server-launch-scenes
```

### 返回数据

```
{
  "sceneList": [{...}],
  "totalCount": 5
}
```

属性	说明
status	返回状态。0：上传成功，返回其它值请参考 <a href="#">API 状态码</a> 。
sceneList	场景列表，同 get-scenes-list

## get-server-settings 接口

获取设备通用设置。

### 请求方式

```
POST http://ip/mwapi/get-server-settings
```

### 返回数据

```
{
  "status": 0,
  "launchScene": 20,
  "autoSwitch": 1,
  "usbMirror": 0,
}
```

属性	说明
status	返回状态。0: 上传成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
launchScene	开机画面场景ID
autoSwitch	连接 HDMI或Webcam源后, 是否自动切换至该源
usbMirror	对 USB-C 输出进行镜像, 使画面内容展示正常

## set-auto-switch 接口

设置连接 HDMI或Webcam源后，自动切换至该源开关。

### 请求方式

```
POST http://ip/mwapi/set-auto-switch
```

```
{  
  "autoSwitch":1  
}
```

属性	说明
autoSwitch	0: 不切换 1: 自动切换至新接入的HDMI或Webcam。

### 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 <a href="#">API 状态码</a> 。

## set-launch-scene 接口

设置当前启用的开机画面。

### 请求方式

```
POST http://ip/mwapi/set-launch-scene
```

```
{  
  "launchScene": 21  
}
```

属性	说明
launchScene	开机画面ID。

### 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 <a href="#">API 状态码</a> 。

## set-usb-mirror 接口

设置 USB Type-C输出画面是否启用镜像。

### 请求方式

```
POST http://ip/mwapi/set-usb-mirror
```

```
{  
  "usbMirror":1  
}
```

属性	说明
usbMirror	0: 不启用镜像 1: 启用镜像

### 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 <a href="#">API 状态码</a> 。

## export-edid 接口

导出 HDMI Source输入端口 EDID, 导出格式为.bin。

### 请求方式

```
POST http://ip/mwapi/export-edid
```

参数	说明
source-id	输入源ID, HDMI 1为0, HDMI 2为1
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 <a href="#">API 状态码</a> 。

# get-edid-config 接口

获取HDMI Source输入端口 EDID 信息。

## 请求方式

POST http://ip/mwapi/get-edid-config

参数	说明
source-id	输入源ID, HDMI 1为0, HDMI 2为1

## 返回数据

```
{
  "status": 0,
  "smart-edid": true,
  "data": "AP////////wA09wEAAQAAAAEaAQOAAAB4Au6Vo1RMmSYPUFT//4AxQEVAyUBxQIGA0QDhwAEAC0gAMPJwWoCwWIoAUB10AAAEAjqAG
HE4LUBYLEUAUB10AAAEAAAA/QAPlg+HPAAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdX19gZWZiY2QHFgMSMgl/B
xUHUD0GwFcGAF9/AWd/AINPAADiAA9uAwwAEAC4eCEQgAECaWn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUQAeMEaPMwBQHxQAA
B4AAAAAAAAAAAAzW=="
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
smart-edid	是否使用 SmartEDID, 有效值: true/false。
data	EDID 信息, base64格式, 显示时需要转换成 16 进制值。

## set-default-edid 接口

重置HDMI Source输入端口 EDID 信息，恢复为默认值。

### 请求方式

POST http://ip/mwapi/set-default-edid

参数	说明
source-id	输入源ID, HDMI 1为0, HDMI 2为1

### 返回数据

```
{
  "status": 0,
  "smart-edid": true,
  "data": "AP////////wA09wEAAQAAAAEaAQOAAAB4Au6Vo1RMmSYPUFT//4AxQEVAYUBxQIGA0QDhwAEAC0gAMPJwWoCwWIoAUB10AAAEAjqAG
HE4LUBYLEUAUB10AAAEAAAA/QAPlg+HPAAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdX19gZWZiY2QHFgMSMgl/B
xUHUD0GwFcGAF9/AWd/AINPAADiAA9uAwwAEAC4eCEQgAECaWn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUQAeMEaPMwBQHxQAA
B4AAAAAAAAAAAAzW=="
}
```

属性	说明
status	返回状态。0: 重置成功，返回其它值请参考 <a href="#">API 状态码</a> 。
smart-edid	SmartEDID, 有效值: true/false。
data	EDID 信息, base64格式, 显示时需要转换成 16 进制值。



## set-edid-config 接口

修改输入端口 EDID 配置信息。

### 请求方式

```
POST http://ip/mwapi/set-edid-config&source-id=0&param1=value1&param2=value2...
```

```
{
  "source-id": 0,
  "smart-edid": true,
  "data": "AP////////wA09wEAAQAAAAEaAQOAAAB4Au6Vo1RMmSYPUFT//4AxQEVAYUBxQIGA0QDhwAEAC0gAMPJwWoCwWIoAUB10AAAEAjqAG
HE4LUBYLEUAUB10AAAEAAAA/QAPlg+HPAAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdX19gZWZiY2QHFgMSMgl/B
xUHUD0GwFcGAF9/AWd/AINPAADiAA9uAwwAEAC4eCEQgAECARn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUQAeMEaPMwBQHxQAA
B4AAAAAAAAAAAAzw=="
}
```

参数	说明
source-id	输入源ID, HDMI 1为0, HDMI 2为1
smart-edid	是否使用SmartEDID, 有效值: true/false。
data	EDID 信息, base64格式, 显示时需要转换成 16 进制值。

### 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 <a href="#">API 状态码</a> 。
source-id	输入源ID, HDMI 1为0, HDMI 2为1

## upload-edid 接口

导入HDMI Source输入端口 EDID 配置信息，导入文件格式为.bin。

### 请求方式

POST http://ip/mwapi/upload-edid

属性	说明
source-id	输入源ID, HDMI 1为0, HDMI 2为1

### 返回数据

```
{
  "status": 0,
  "data": "AP////////wA09wEAAQAAAAEaAQOAAAB4Au6Vo1RMmSYPUFT//4AxQEVAyUBxQIGA0QDhwAEAC0gAMPJwWoCwWIoAUB10AAAEAjqAG
HE4LUBYLEUAUB10AAAEAAAA/QAP1g+HPAAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdX19gZWZiY2QHFgMSMgl/B
xUHUD0GwFcGAF9/AWd/AINPAADiAA9uAwwAEAC4eCEQgAECARn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUAeMEaPMwBQHxQAA
B4AAAAAAAAAAAAzW=="
}
```

字段	说明
status	返回状态。0: 导入成功，返回其它值请参考 <a href="#">API 状态码</a> 。
data	导入的 EDID 信息，base64格式，显示时需要转换成 16 进制值。

## get-signal-info 接口

获取HDMI Source输入信号状态信息。

### 请求方式

```
POST http://ip/mwapi/get-signal-info/
```

参数	说明
source-id	输入源ID, HDMI 1为0, HDMI 2为1

### 返回数据

JSON 结构如下:

```
{
  "status": 0,
  "signal-info-types": ["video-info", "audio-info", "hdmi-info", "info-frames"],
  "video-info": {...},
  "audio-info": {...},
  "hdmi-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 <a href="#">API 状态码</a> 。

### 信号类型

```
"signal-info-types": [
  "video-info",
  "audio-info",
  "hdmi-info",
  "info-frames"
]
```

### 视频信号状态 (video-info: {...})

```
"video-info": {
  "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
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	帧率, 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": "hdmi",
  "hdcv": "none",
  "vic": 0,
  "it-content": false,
  "pixel-rate": "148MHz",
  "timing-htotal": 2200,
  "timing-hactive": 1920,
  "timing-h-front-porch": 88,
  "timing-h-synct-width": 44,
  "timing-h-back-porch": 148,
  "timing-vtotal": 1125,
  "timing-vactive": 1080,
  "timing-v-front-porch": 4,
  "timing-v-synct-width": 5,
  "timing-v-back-porch": 36
}
```

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。
hdcv	HDCP 加密方式, 有效值: none、hdcv-1.x、hdcv-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。

#### 信息帧 (info-frames: {...})

```

"info-frames": [
  {
    "id": "AVI",
    "type": 130,
    "version": 2,
    "length": 13,
    "checksum": 96,
    "data": "ACgAIgAAADkEAACBBw=="
  },
  {
    "id": "SPD",
    "type": 132,
    "version": 1,
    "length": 25, //bytes
    "checksum": 112,
    "data": "AQAAAAAAAAAAAA=="
  }
]

```

属性	说明
id	信息帧名称, 包括 AVI、SPD...
type	报文类型。
version	版本。
length	长度。
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HDMI Source视频设置信息。

## 请求方式

POST http://ip/mwapi/get-video-config

参数	说明
source-id	输入源ID, HDMI 1为0, HDMI 2为1

## 返回数据

```
{
  "status": 0,
  "in-auto-color-fmt": true,
  "in-color-fmt": "rgb",
  "in-auto-quant-range": true,
  "in-quant-range": "full",
  "brightness": 0,
  "contrast": 100,
  "hue": 0,
  "saturation": 100,
  "deinterlace": "none",
  "out-mirror": false,
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 <a href="#">API 状态码</a> 。
in-auto-color-fmt	输入色彩空间是否自动获取，有效值：true/false。
in-color-fmt	输入色彩空间值，有效值：rgb、bt.601、bt.709、bt.2020。
in-auto-quant-range	输入量化范围是否自动获取，有效值：true/false。
in-quant-range	输入量化范围值，有效值：full、limited。
brightness	亮度，有效值：-100 ~ +100。
contrast	对比度，有效值：50 ~ 200。
hue	色调，有效值：-90 ~ 90。
saturation	饱和度，有效值：0 ~ 200。
deinterlace	去隔行，有效值：none、top-field、bottom-field。
out-mirror	是否镜像，有效值：true/false。

# set-video-config 接口

修改HDMI Source视频配置信息。

## 请求方式

POST http://ip/mwapi/set-video-config

参数	说明
source-id	输入源ID, HDMI 1为0, HDMI 2为1

## 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 <a href="#">API 状态码</a> 。
source-id	输入源ID, HDMI 1为0, HDMI 2为1

## 接口示例

### 1. 输入色彩空间设置

```
{
  "in-auto-color-fmt": false,
  "in-color-fmt": 0
}
```

参数	说明
in-auto-color-fmt	输入色彩空间是否自动获取, 修改时值必须为 false。
in-color-fmt	输入色彩空间值, 有效值: rgb、bt.601、bt.709、bt.2020。

### 2. 输入量化范围设置

```
{
  "in-auto-quant-range": false,
  "in-quant-range": "full"
}
```

参数	说明
in-auto-quant-range	输入量化范围是否自动获取, 修改时值必须为 false。
in-quant-range	输入量化范围值, 有效值: full、limited。

### 3. 色彩处理设置

```
{
  "brightness": 0,
  "contrast": 100,
  "hue": 0,
  "saturation": 12
}
```

参数	说明
brightness	亮度, 整数, 取值范围: -100 ~ +100。

contrast	对比度, 整数, 取值范围: 50 ~ 200。
hue	色调, 整数, 取值范围: -90 ~ 90。
saturation	饱和度, 整数, 取值范围: 0 ~ 200。

#### 4. 去隔行设置

```
{
  "deinterlace":"none"
}
```

参数	说明
deinterlace	去隔行, 有效值: none、top-field、bottom-field。

#### 4. 特殊效果设置

```
{
  "out-mirror":true
}
```

参数	说明
out-mirror	镜像, 有效值: true、false。



## get-webcam-config 接口

获取webcam Source视频设置信息。

### 请求方式

POST http://ip/mwapi/get-webcam-config

### 返回数据

```
{
  "status": 0,
  "device-name": "Logical HD Camera",
  "formats": [
    {
      "fourcc": "NV12",
      "resolutions": [
        {
          "width": 1920,
          "height": 1080,
          "fps": 6000
        },
        {
          "width": 1280,
          "height": 720,
          "fps": 3000
        }
      ]
    }
  ]
  "out-mirror": false,
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 <a href="#">API 状态码</a> 。
device-name	设备名称。
formats	格式列表。
out-mirror	是否镜像，有效值：true/false。

## set-webcam-config 接口

修改webcam Source视频配置信息。

### 请求方式

POST http://ip/mwapi/set-webcam-config

```
{
  "width": 1920,
  "height": 1080,
  "fps": 6000,
  "fourcc": "MJPG",
  "out-mirror": true
}
```

属性	说明
width	采集宽度。
height	采集高度。
fps	采集帧率 (x100) 。
fourcc	采集格式。
out-mirror	是否镜像, 有效值: true/false。

### 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 <a href="#">API 状态码</a> 。

# get-note-thumbnail 接口

获取笔记场景缩图

## 请求方式

POST http://ip/mwapi/get-note-thumbnail

参数	说明
method	方法名称: delete-scene
sceneId	笔记场景ID
isThumbnail	是否为缩略图 0:suoluetu 1:原图
modificationTime	笔记编辑时间戳

```
{
  "sceneId":102,
  "isThumbnail":0,
  "modificationTime":0
}
```

## 返回数据

JSON 结构如下:

```
{
  "info":{
    "isThumbnail":0,
    "sceneId":110,
    "thumbnail":{
      "data":"",
      "encoding":"base64",
      "modificationTime":8953299
    }
  },
  "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 <a href="#">API 状态码</a> 。
sceneIds	笔记场景ID
isThumbnail	是否为缩略图 0:suoluetu 1:原图
modificationTime	笔记编辑时间戳
encoding	data加密方式, 此处使用base64
data	加密后的数据内容, 注意: 请求中modificationTime已是最新时, data内容为空, 减少网络传输

## HTTP访问方式

笔记缩略图:[http://\[IP\]/note/\[演示稿ID\]/\[笔记ID\]/0](http://[IP]/note/[演示稿ID]/[笔记ID]/0)

笔记预览图:[http://\[IP\]/note/\[演示稿ID\]/\[笔记ID\]/1](http://[IP]/note/[演示稿ID]/[笔记ID]/1)

## export-output-edid 接口

导出 HDMI Output输出端口 EDID, 导出格式为.bin。

### 请求方式

```
POST http://ip/mwapi/export-output-edid
```

参数	说明
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 <a href="#">API 状态码</a> 。

# get-hdmi-output-info 接口

获取HDMI Output 信息。

## 请求方式

POST http://ip/mwapi/get-hdmi-output-info

## 返回数据

```
{
  "status": 0,
  "mode": 0,
  "data": "AP////////wA09wEAAQAAAAEaAQ0AAAB4Au6Vo1RMmSYPUFT//4AxQEVAyUBxQIGA0QDhwAEAC0gAMPJwWoCwWIoAUB10AAAEAjqAG
HE4LUBYLEUAUB10AAAEAAAA/QAPlg+HPAAAAAAAAAAAAAAAA/ABNQUdFV0VMTAogICAgAWYCA1HxV2EQHwQTBRQgISJdX19gZWZiY2QHFgMSMgl/B
xUHUD0GwFcGAF9/AWd/AINPAADiAA9uAwwAEAC4eCEQgAECaWn2F3EAXiAA+MPAeABHYAYcRwWIFgsJQBAhGMAAJ5mIVaqUQAeMEaPMwBQHxQAA
B4AAAAAAAAAAAAzw=="
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
mode	mode 类型, 0: LOOPTHRU HDMI 1, 1: LOOPTHRU HDMI 2, 2: PGM。
data	EDID 信息, base64格式, 显示时需要转换成 16 进制值。

## set-hdmi-output-mode 接口

设置hdmi 输出模式。

### 请求方式

```
POST http://ip/mwapi/set-hdmi-output-mode
```

```
{  
  "mode": 0  
}
```

参数	说明
mode	mode 类型, 0: LOOPTHRU HDMI 1, 1: LOOPTHRU HDMI2, 2: PGM。

### 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 <a href="#">API 状态码</a> 。

# get-usb-output-config 接口

获取usb output信息。

## 请求方式

POST http://ip/mwapi?method=get-usb-output-config

## 返回数据

```
{
  "status": 0,
  "uvc-formats": [
    {
      "fourcc": "NV12",
      "resolutions": [
        {
          "width": 1920,
          "height": 1080,
          "fps": 6000
        },
        {
          "width": 1280,
          "height": 720,
          "fps": 3000
        }
      ]
    }
  ],
  "uac-formats": [
    {
      "sample-rate": 48000,
      "channels": 2,
      "bits-count": 16
    }
  ]
}
```

属性	说明
status	返回状态。0：数据获取成功，返回其它值请参考 <a href="#">API 状态码</a> 。
uvc-formats	UVC格式列表。
uac-formats	UAC格式列表。

# add-presentation 接口

新建演示稿

## 请求方式

POST http://ip/mwapi/add-presentation

参数	说明
name	演示稿名称

```
{
  "name": "New Presentation 1"
}
```

## 返回数据

JSON 结构如下:

```
{
  "message": "success",
  "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 <a href="#">API 状态码</a> 。



# delete-presentation 接口

删除演示稿

## 请求方式

```
POST http://ip/mwapi/delete-presentation
```

参数	说明
id	演示稿ID

```
{  
  "id":8  
}
```

## 返回数据

JSON 结构如下:

```
{  
  "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 <a href="#">API 状态码</a> 。

## get-presentations-list 接口

获取演示稿列表

### 请求方式

POST http://ip/mwapi/get-presentations-list

### 返回数据

JSON 结构如下:

```
{
  "status":0,
  "presentationsList":[
    {
      "id":2,
      "name":"新建文稿2",
      "isCurrent":1,
      "sceneList":[]
    },
    {
      "id":3,
      "name":"新建文稿3",
      "isCurrent":0,
      "sceneList":[]
    }
  ]
}
```

属性	说明
status	返回状态。0: 数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
id	演示稿ID
name	名称
isCurrent	是否为当前演示稿 0:非 1:是
sceneList	演示稿包含的场景列表

# rename-presentation 接口

重命名演示稿

## 请求方式

POST http://ip/mwapi/rename-presentation

参数	说明
id	演示稿ID
name	演示稿名称

```
{
  "id":8,
  "name": "新建演示稿1"
}
```

## 返回数据

JSON 结构如下:

```
{
  "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 <a href="#">API 状态码</a> 。

# switch-presentation 接口

切换演示稿

## 请求方式

POST http://ip/mwapi/switch-presentation

参数	说明
id	演示稿ID

```
{  
  "id":1  
}
```

## 返回数据

JSON 结构如下:

```
{  
  "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 <a href="#">API 状态码</a> 。

## check-source-exist 接口

检查资源文件是否已存在

### 请求方式

POST http://ip/mwapi/check-source-exist

参数	说明
sourceType	资源类型 1: 图片 2: 视频 6: 音乐
md5	资源文件md5值

### 返回数据

JSON 结构如下:

```
{
  "info": {
    "find": 1,
    "source": {
      "duration": 0,
      "filePath": "2009/2009.JPG",
      "height": 1080,
      "id": 2009,
      "md5": "f7e7f0cd578a44c77aed69e7c147d676",
      "name": "IMG_3010",
      "rotation": 0,
      "secondaryType": 0,
      "thumbHeight": 144,
      "thumbWidth": 256,
      "thumbnailPath": "2009/2009.thumb.jpeg",
      "type": 1,
      "width": 1920
    }
  },
  "message": "success",
  "result": 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 <a href="#">API 状态码</a> 。
find	0:未找到 1:找到指定预置资源
source	资源实体

## check-source-used 接口

检查资源文件是否被使用

### 请求方式

POST http://ip/mwapi/check-source-used

参数	说明
sourceType	资源类型 1: 图片 2: 视频 6: 音乐
id	资源ID

### 返回数据

JSON 结构如下:

```
{
  "result":0,
  "message":"success",
  "info":{
    "used":1
  }
}
```

属性	说明
status	返回状态。0:数据获取成功, 返回其它值请参考 <a href="#">API 状态码</a> 。
used	0:未被使用 1:被某演示稿使用 2:被当前预览画面使用,不可删除

# delete-source 接口

删除资源

## 请求方式

```
POST http://ip/mwapi/delete-source
```

参数	说明
id	资源ID
sourceType	资源类型

## 返回数据

JSON 结构如下:

```
{  "message": "The resource is previewing",  "status": 40}
```

### 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 <a href="#">API 状态码</a> 。

# get-sources-list 接口

获取服务器资源列表

## 请求方式

POST http://ip/mwapi/get-sources-list

参数	说明
type	资源类型。1: 图片 2: 视频 6: 音乐 0: 预置资源 (采集+图片+视频)

## 返回数据

JSON 结构如下:

```
{
  "info": {
    "pageIndex": 0,
    "sourceList": [
      {
        "duration": 0,
        "filePath": "4/4.jpeg",
        "height": 1461,
        "id": 4,
        "name": "默认图片",
        "thumbnailPath": "4/4.thumb.jpeg",
        "type": 1,
        "width": 2224
      },
      {
        "duration": 0,
        "filePath": "100/100.jpeg",
        "height": 1080,
        "id": 100,
        "name": "ubuntu",
        "thumbnailPath": "100/100.thumb.jpeg",
        "type": 1,
        "width": 1920
      },
      {
        "duration": 120000,
        "filePath": "101/101.mp4",
        "height": 1080,
        "id": 101,
        "name": "1080p-red",
        "thumbnailPath": "101/101.thumb.jpeg",
        "type": 2,
        "width": 1920
      },
      {
        "duration": 0,
        "filePath": "1018/1018.JPG",
        "height": 720,
        "id": 1018,
        "name": "IMG_2878",
        "thumbnailPath": "1018/1018.thumb.jpeg",
        "type": 1,
        "width": 1280
      },
      {
        "duration": 1234,
        "filePath": "1041/1041.mp3",
        "height": 0,
        "id": 1041,
        "name": "IMG_2969",
        "thumbnailPath": "1041/1041.thumb.jpeg",
      }
    ]
  }
}
```



```

        "type":6,
        "width":0,
        "artist":"Magewell"
    }
],
"totalCount":11
},
"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 <a href="#">API 状态码</a> 。
id	资源ID
type	资源类型。1：图片；2：视频；3：HDMI 1；4：HDMI 2；5：USB CAMERA；6：音乐；7：笔记背景图
name	资源名称
thumbnailPath	资源缩略图相对路径。可通过http访问 ( <a href="#">http://设备IP/file/相对路径</a> )
filePath	资源文件相对路径。可通过http访问 ( <a href="#">http://设备IP/file/相对路径</a> )
width	资源宽度。除音乐类型外有效
height	资源高度。除音乐类型外有效
duration	资源持续时间。仅视频、音乐有效
artist	作者。仅音乐有效

# import-source 接口

导入资源

## 请求方式

POST http://ip/mwapi/import-source

参数	说明
fileName	文件名称
md5	文件md5值
sourceType	资源类型 1: 图片 2: 视频 6: 音乐

ps:文件已通过"/mwapi/upload-source-file"上传至server端设备

## 返回数据

JSON 结构如下:

```
{
  "info":{
    "duration":0,
    "filePath":"2039/2039.png",
    "height":1080,
    "id":2039,
    "name":"IMG_3162",
    "rotation":0,
    "thumbnailPath":"2039/2039.thumb.jpeg",
    "type":1,
    "width":1440
  },
  "message":"success",
  "result":0
}
```

### 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 <a href="#">API 状态码</a> 。
info	资源实体

# rename-source 接口

重命名资源

## 请求方式

POST http://ip/mwapi/rename-source

参数	说明
id	资源ID
name	新名称

```
{
  "id": 2039,
  "name": "New Source Name"
}
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。

# upload-source-file 接口

资源文件上传

## 请求方式

```
POST http://ip/mwapi/upload-source-file
```

## 返回数据

JSON 结构如下:

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

### 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 <a href="#">API 状态码</a> 。