

Magewell Cloud

前言

文档简介	1.1
接口约定	1.2
设备类型	1.3
API状态码	1.4
异常设备状态掩码	1.5
DEMO (Node.js)	1.6

用户登录

login	2.1
logout	2.2

通用接口

get-status	3.1
get-dev-catalogs	3.2
get-all-firmwares	3.3
get-logs	3.4
clear-logs	3.5

待审核

get-wait-devs	4.1
get-reject-devs	4.2
accept-dev	4.3
reject-dev	4.4
delete-dev	4.5

所有设备

get-devs	5.1
get-warning-devs	5.2
reboot	5.3
rename	5.4
start-upgrade	5.5
start-rec	5.6
stop-rec	5.7
start-live	5.8
stop-live	5.9
get-dev	5.10
set-dev-secu	5.11
get-dev-logs	5.12
WebGUI工具	5.13

SRT服务器

get-srt-server	6.1
set-srt-settings	6.2
get-srt-server-settings	6.3
start-srt-server	6.4
stop-srt-server	6.5

协议转换

get-channels	7.1
add-channel	7.2
set-channel-source	7.3
set-channel-name	7.4
set-channel-enable	7.5
delete-channel	7.6
add-channel-sink	7.7
set-channel-sink	7.8
set-channel-sink-name	7.9
set-channel-sink-enable	7.10
delete-channel-sink	7.11

安全设置

set-code	8.1
set-web	8.2
uploadCert	8.3
uploadCertKey	8.4

授权管理

activation	9.1
deactivation	9.2

设备固件管理

uploadFirmware	10.1
add-firmware	10.2
delete-firmware	10.3

用户管理

get-users	11.1
add-user	11.2
del-user	11.3
change-passwd	11.4
reset-passwd	11.5

设备分组

add-catalog	12.1
rename-catalog	12.2
delete-catalog	12.3
set-catalog	12.4

文档简介

针对Magewell Cloud，我们开放了丰富的API，方便开发人员自定义工作流程，修改云平台配置等。这些API基于HTTP/HTTPS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。

接口约定

概要

- 请求协议：HTTP/HTTPS。
- 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[:port]/xxx?param1=value1¶m2=value2....
- 返回数据格式：HTTP状态为200时，返回JSON数据，否则为HTTP对应错误。
- 登录认证方式：在Cookie中携带mwcloud-sid=xxxxxxxx和mwcloud-uid=xxxxxxxx。

请求响应

请求响应格式为JSON数据。

其中result是请求的响应码。

- 取值为0时，表示数据获取或操作成功。
- 其他参见[API状态码](#)。

```
{
  "remote-ip": "127.0.0.1:33172",
  "result": 0,
  "status": {
    "system-info": {
      "date-time": 1634882136,
      "cloud-version": "1.1.22535"
    }
  }
  ...
}
```

设备类型

支持云平台注册的设备类型。

产品族

产品族编号	说明
0	所有产品族
1	Ultra Stream
2	Ultra Encode
3	Pro Convert编码器
4	Pro Convert解码器

产品列表

产品编号为十六进制数字。

产品编号	产品族编号	产品型号
0x301	1	Ultra Stream HDMI
0x302	1	Ultra Stream SDI
0x304	2	Ultra Encode HDMI
0x305	2	Ultra Encode SDI
0x401	3	Pro Convert HDMI 4K Plus
0x402	3	Pro Convert HDMI Plus
0x403	3	Pro Convert SDI 4K Plus
0x404	3	Pro Convert SDI Plus
0x405	3	Pro Convert HDMI TX
0x406	3	Pro Convert SDI TX
0x409	3	Pro Convert 12G SDI 4K Plus
0x410	4	Pro Convert for NDI to HDMI
0x411	4	Pro Convert for NDI to HDMI 4K
0x413	4	Pro Convert H.26x to HDMI
0x414	4	Pro Convert H.26x to HDMI 4K
0x415	4	Pro Convert for NDI to SDI
0x417	4	Pro Convert H.26x to SDI
0x421	4	Pro Convert for NDI to AIO

API状态码

```

{
    SC_RET_INVALID_PARAMETER          int = 10000
    SC_RET_INVALID_COMMAND            int = 10001
    SC_RET_INVALID_REQUEST            int = 10002
    SC_RET_INVALID_PORT_NUMBER        int = 10003
    SC_RET_CONNECT_TARGET_TIMEOUT     int = 10004
    SC_RET_HTTP_METHOD_NOT_SUPPORTED int = 10005
    SC_RET_CLIENT_NOT_EXIST           int = 10006
    SC_RET_CLIENT_OFFLINE             int = 10007
    SC_RET_NEEDS_AUTH                 int = 10008

    retUpgradeCompleted int = 109
    retUpgradeIdel      int = 108
    retUpgradeFailed    int = 107
    retUpgradeUpdating  int = 106

    retSamePort          int = 105 // 端口被占用
    retDeleted           int = 104 // 云平台上删除了此设备
    retWaiting           int = 103 // 正注册设备到云平台上...
    retRefused           int = 102 // 审核设备时, 拒绝
    retAccepted          int = 101 // 审核设备时, 接受
    retLivingDNS         int = 28
    retLivingInit        int = 27
    retReadyForRecord    int = 26
    retBitrate           int = 25
    retLivingWaiting     int = 24
    retLivingConnecting  int = 23
    retLivingConnected   int = 22
    retPushReboot        int = 21
    retAudioSignalChange int = 20
    retBlueWrite         int = 19
    retBlueRead          int = 18
    retBlueShutDown      int = 17
    retDiskOn            int = 16
    retDiskOff           int = 15
    retDiskChange        int = 14
    retSnapshotOver      int = 13
    retPushReset         int = 12
    retPushLiving        int = 11
    retPushRecord        int = 10
    retSignalChange      int = 9
    retRouteChange       int = 8
    retIPChange          int = 7
    retNetChange         int = 6
    retExit              int = 5
    retLowSpace          int = 4
    retLowSpeed          int = 3
    retRunning           int = 2
    retRepeat            int = 1 // 重复操作
    retSucceed           int = 0 // 请求成功
    errPasswd           int = -1
    errOccupied         int = -2
    errDisconnect       int = -3
    errDevice           int = -4
    errDisk             int = -5
    errUnconnect        int = -6
    errKey              int = -7
    errVersion          int = -8
    errBusy             int = -9
    errParam            int = -10
    errUsage            int = -11
    errTimeout          int = -12
    errIP               int = -13
    errNotFound         int = -14
    errFile             int = -15
    errNoSpace          int = -16
    errNeedAuth         int = -17
    errSystem           int = -18
    errDiskSpeed        int = -19
    errEmpty            int = -20
    errNetwork          int = -21
    errEvent            int = -22

```



```
errCodec          int = -23
errBlue           int = -24
errDB             int = -101
errCALicense      int = -102
errCAKey          int = -103
errNoLogin        int = -104 // 登录失效
errValues         int = -105
errFullLicense    int = -106
errOverTime       int = -107 // 云平台证书授权有效期已过
errClientID       int = -108
}
```

异常设备状态掩码

```
{  
  warningNoSignal    = 0x01 // 无信号  
  warningNoUSB       = 0x02 // U盘无效  
  warningFullUSB     = 0x04 // U盘已满  
  warningFullMSD     = 0x08 // 本地存储空间已满  
  warningNoSD        = 0x10 // 本地存储空间无效  
}
```

DEMO (Node.js)

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

DEMO下载链接: [magewell-cloud-api-demo-nodejs.zip](https://github.com/magewell-cloud-api-demo-nodejs)。

DEMO目录结构:

```
magewell-cloud-api-demo-nodejs
|
|-- httpUtils.js // 基于nodejs的HTTP模块封装了get、post和upload方法
|-- constant.js // 常量
|-- xxxx.mwf // upload.js调用uploadFirmware接口上传设备的测试固件
|-- get.js // 通过GET方式调用接口获取数据
|-- upload.js // 通过POST方式实现文件上传
```

环境准备

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

运行方式

1. 在终端控制台进入DEMO目录。

```
cd magewell-cloud-api-demo-nodejs
```

2. 将constant.js中的CLOUD_IP替换成云平台的IP地址。
3. 运行get.js。

```
node get
```

4. 运行upload.js。

```
node upload
```

login接口

用户登录接口。

登录成功后，会在Cookie中存放Session ID(Cookie:mwcloud-sid=xxx;mwcloud-uid=xxx)。

接口原型

协议	HTTP/HTTPS POST
URL	ip:[port]/sc_cloud/login
数据格式	请求消息: Content-Type: multipart/form-data 响应消息: Content-Type: application/json

请求参数

Request Header

参数	描述
Content-Type	取值为: 'multipart/form-data;'

Request Body

参数	描述
id	用户名。
pass	密码，需MD5加密。

请求示例

```
POST /sc_cloud/login HTTP/1.1
Content-Length: 261
Accept: application/json, text/plain, */*
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryGXBvCk8Vzymbt6WUB
-----4437964479282813733506153675
Content-Disposition: form-data; name="id"

Admin
-----4437964479282813733506153675
Content-Disposition: form-data; name="pass"

e3afed0047b08059d0fada10f400c1e5
-----4437964479282813733506153675--
```

响应Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
id	用户ID。
name	用户名称。
result	返回码。 <ul style="list-style-type: none">• 0: 登录成功。• -1: 密码错误。• -14: 用户不存在。• 其它: 参见API状态码。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 85
{
  "id":1,
  "name":"Admin",
  "pass":"",
  "result":0,
  "sid":"OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP"
}
```

logout接口

退出登录，返回到登录界面。

接口原型

协议	HTTP/HTTPS POST
URL	ip:[port]/sc_cloud/logout
数据格式	响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
POST /sc_cloud/logout HTTP/1.1
Content-Length: 0
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 退出登录成功。其它: 参见API状态码。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 53
Set-Cookie: mwcloud-sid=; Path=/; Domain=127.0.0.1; Max-Age=60; HttpOnly
{
  "result":0,
  "sid":"OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP"
}
```

get-status接口

实时获取云平台的基本信息，主要有CPU利用率、内存使用量、网络吞吐量、平台安全设置、异常设备信息、日志等。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-status
数据格式	响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-status HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
status	云平台状态信息。

status

参数	描述
system-info	云平台系统信息。
system-status	云平台系统性能。
app-settings	云平台安全设置。
all-count	审核通过的设备总数。
warning-count	异常设备总数。
pending-count	等待审核的设备总数。
online-count	在线设备总数。
cur-convert-count	已配置的协议流转换任务总数。
cur-srt-count	已配置的SRT输入输出流总数。
license-reg	注册到云平台的授权证书。
cur-license	云平台当前使用的授权证书。
warning-devs	最新上报警告的设备。所有有警告的设备列表通过 get-warning-devs 接口获取。
dev-logs	最新上报的设备日志。获取指定设备的运行日志参见 get-dev-logs 接口。
sys-logs	部分云平台日志。获取云平台的运行日志参见 get-logs 接口。

system-info

参数	描述
date-time	当前系统时间，单位：秒。
cloud-version	系统版本号。

system-status

参数	描述
cpu	CPU利用率。
mem	内存使用量。
net-tx	网络发送吞吐量。
net-rx	网络接收吞吐量。

app-settings

参数	描述
is-code	设备注册到云平台是否需要邀请码。0：否 1：是。
code-value	4位数邀请码。NULL：无邀请码。
is-http	是否允许HTTP方式访问云平台。0：否 1：是。
http-port	端口。
is-https	是否允许HTTPS方式访问云平台。0：否 1：是。
https-port	端口。
is-cert-valid	CA证书是否有效。0：无效 1：有效。
is-certkey-valid	CA私钥是否有效。0：无效 1：有效。

license-reg

参数	描述
status	授权证书的状态码。取值如下： <ul style="list-style-type: none"> 0：添加的授权证书有效。 -20：未添加授权证书，则使用云平台默认的证书。 -107：授权证书已过期。 -10：证书错误。
user-name	证书标题。
sn	证书序列号。
license	已授权证书的权限信息。

license

参数	描述
max-dev-count	最多允许管理的设备个数。
max-convert-count	最多允许添加的协议流转换任务个数。
max-srt-count	最多允许添加的SRT输入输出流个数。
limited-date	授权有效期，单位：秒。

warning-devs

参数	描述
name	设备名称。
SN	设备序列号。
module-name	设备产品型号。
hardware-ver	设备硬件版本号。
firmware-ver	设备固件版本号。
dev-warning	是否是警告设备。0：不是； 1：是。
is-online	是否在线。

logs

参数	描述
id	日志ID。
date	日志产生时间，单位：秒。
info	日志信息。
type	日志类型。1：信息，2：警告，3：错误。
sn	设备序列号。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "result": 0,
  "status": {
    "system-info": {
      "date-time": 1634891748,
      "cloud-version": "1.1.1"
    },
    "system-status": {
      "cpu": [],
      "mem": [],
      "net-tx": [],
      "net-rx": []
    },
    "app-settings": {
      "is-https": 1,
      "https-port": 443,
      "is-cert-valid": 1,
      "is-certkey-valid": 1,
      "is-http": 1,
      "http-port": 80,
      "is-code": 0,
      "code-value": "NULL"
    },
    "all-count": 18,
    "warning-count": 13,
    "pending-count": 0,
    "online-count": 7,
    "cur-convert-count": 7,
    "cur-srt-count": 3,
    "license-reg": {
      "status": 0,
      "user-name": "magewell-ltd",
      "sn": "9C4E*****F769",
      "license": {
        "max-dev-count": 100,
        "max-convert-count": 3,
        "max-srt-count": 2,
        "limited-date": 1769904000
      }
    },
    "cur-license": {
      "max-dev-count": 100,
      "max-convert-count": 3,
      "max-srt-count": 2,
      "limited-date": 1769904000
    },
    "warning-devs": [
      {
        "name": "Ultra Stream",
        "SN": "xxx",
        "module-name": "Ultra Stream SDI",
        "hardware-ver": "A",
        "firmware-ver": "1.0.0",
        "dev-warning": 1,
        "is-online": true
      }
    ],
    "dev-logs": [
      {
        "id": 3366,
        "date": 1634891480,
        "info": "Offline.",
        "type": 2,
        "sn": "xxx"
      }
    ],
    "sys-logs": []
  }
}
```

get-dev-catalogs接口

获取用户在云平台上自定义的设备分组。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-dev-catalogs
数据格式	响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-dev-catalogs HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
dev-catalogs	设备分组集合。

dev-catalogs

设备分组参数解析。

参数	描述
id	设备分组ID。
name	分组名称。
is-everyone	是否所有用户可见。0: 否, 1: 是。
users	is-everyone取值为0时有效, 可见用户集合。
childs	子分组集合。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 310
{
  "result": 0,
  "dev-catalogs": [
    {
      "id": 1,
      "name": "Group1",
      "is-everyone": 1,
      "users": null,
      "childs": [
        {
          "id": 3,
          "name": "Group11",
          "is-everyone": 1,
          "users": null,
          "childs": [
            {
              "id": 4,
              "name": "Group111",
              "is-everyone": 0,
              "users": [
                "Admin",
                "Test"
              ],
              "childs": null
            }
          ]
        }
      ]
    }
  ],
  {
    "id": 2,
    "name": "Group2",
    "is-everyone": 0,
    "users": [
      "Admin"
    ],
    "childs": null
  }
]
}
```

get-all-firmwares接口

获取云平台上指定设备的所有可用固件。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-all-firmwares?dev-type=1&product-id=770&hardware-ver=A
数据格式	响应消息: Content-Type: application/json

请求URL参数

参数	是否必选	描述
dev-type	是	设备产品族编号。
product-id	否	设备产品编号。
hardware-ver	否	设备硬件版本。

当请求URL参数只有dev-type, 且dev-type=0时, 响应数据为云平台上所有可用的固件。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-all-firmwares?dev-type=1&product-id=770&hardware-ver=A HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
firmwares	指定设备的所有可用固件。

firmwares

指定设备的所有可用固件。

参数	描述
id	固件ID。
file-path	固件存放路径。
upload-date	固件上传时间。
product-id	设备产品编号。
firmware-id	固件版本类型。 <ul style="list-style-type: none"> 取值为1，测试版本。 其他为发布版本。
hardware-ver	设备硬件版本。
marjor-ver	主版本号。
minor-ver	次版本号。
build-number	构建号。

设备固件版本由marjor-ver、minor-ver和build-number组成。
如响应示例中固件版本为1.5.178。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 219
{
  "result": 0,
  "firmwares": [
    {
      "id": 21,
      "file-path": "/data/firmware/firmware-770-1-A-1.5.178.mwf",
      "upload-date": 1635151650,
      "product-id": 770,
      "firmware-id": 1,
      "hardware-ver": "A",
      "marjor-ver": 1,
      "minor-ver": 5,
      "build-number": 178
    }
  ]
}
```

get-logs接口

获取云平台的运行日志。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-logs
数据格式	响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-logs HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
logs	云平台日志列表。

logs

参数	描述
id	日志ID。
date	日志产生时间。
info	日志信息。
type	日志类型。1: 信息, 2: 警告, 3: 错误。
sn	云平台序列号。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "result": 0,
  "logs": [
    {
      "id": 4030,
      "date": 1635479011,
      "info": "Admin logged in from '10.10.13.100'.",
      "type": 1,
      "sn": "CLOD200914001"
    }
  ]
}
```


clear-logs接口

清除云平台或指定设备的运行日志。

云平台日志清除只有管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/log/clear
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
sn	序号。 <ul style="list-style-type: none">• CLOD200914001: 清除云平台日志。• 其他: 清除序号对应的设备日志。

请求示例

```
POST /sc_cloud/log/clear HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 22
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "sn": "CLOD200914001"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
method	方法名: clear-logs。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 34
{
  "method": "clear-logs",
  "result": 0
}
```

get-wait-devs接口

获取当前申请加入云平台的设备列表。

仅管理员有权限。

针对待审核设备，可以进行以下操作：

- [审核通过](#)
- [审核拒绝](#)
- [删除设备](#)

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-wait-devs
数据格式	响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-wait-devs HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
wait-devs	待审核设备列表。

wait-devs

参数	描述
catalog-id	设备分组ID。
sn	序列号。
module	产品族编号。
dev-warning	取值大于0，表示异常设备。 dev-warning取值与异常设备状态掩码中的所有掩码做逻辑与计算，可能有多个异常。 例如：(dev-warning & 0x01)=== 0x01，表示此设备无信号。
is-everyone	是否所有用户可见。0：否，1：是。
users	is-everyone取值为0时有效，可见用户集合。
apply-date	申请时间。
online-date	设备在线时间。大于0：设备在线，小于等于0：设备离线。
is-upload-file	是否在上传固件。1：正在上传固件到设备上。 使用参见： 设备升级流程 。
upload-file-pos	固件上传进度百分比。
status	设备的基本信息。

status

参数	描述
name	设备名称。
product-id	设备产品编号。
module-name	设备产品型号。
hardware-ver	硬件版本。
firmware-ver	固件版本。
up-time	设备上电时间，单位：秒。
cur-status	设备状态掩码，不同设备不同处理。
eth	以太网IP。
wifi	WI-FI IP。
rndis	USB RNDIS IP。
upgrade-ret	设备安装固件状态码。参见： API状态码 。 使用参见： 设备升级流程 。
upgrade-step	当前升级步骤，一共有4步。
upgrade-percent	当前设备升级进度。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "result": 0,
  "wait-devs": [
    {
      "catalog-id": 0,
      "sn": "B401190213050",
      "module": 3,
      "dev-status": 0,
      "network": "",
      "is-upgrade": 0,
      "users": null,
      "is-everyone": 0,
      "apply-date": 1635152916,
      "online-date": 1635152917,
      "offline-date": 0,
      "cpu": [],
      "mem": [],
      "net": [],
      "net2": [],
      "net3": [],
      "sid": "7e0d83332a03a5074268b19286063722",
      "status": {
        "name": "Pro Convert",
        "product-id": 1025,
        "module-name": "HDMI 4K Plus",
        "hardware-ver": "B",
        "firmware-ver": "1.1.207",
        "conn-ip": "",
        "up-time": 30491,
        "eth": "10.10.8.13",
        "sd-total": 0,
        "cur-status": 0,
        "wifi": "",
        "input": "1920x1080p60.00",
        "output": "unconnected",
        "usb-total": 0,
        "usb-free": 0,
        "usb-used": 0,
        "usb-status": 0,
        "usb-usage": 0,
        "sd-free": 0,
        "sd-used": 0,
        "sd-status": 0,
        "sd-usage": 0,
        "rec-codec": "",
        "rec-sec": 0,
        "rec-kbps": 0,
        "live-codec": "",
        "live-sec": 0,
        "live-kbps": 0,
        "ndi-connected": false,
        "ndi-url": "",
        "ndi-enabled": true,
        "rndis": "192.168.65.1",
        "ndi-name": "#00 B401190213050",
        "ndi-video-width": 1920,
        "ndi-video-height": 1080,
        "ndi-video-scan": "progressive",
        "ndi-video-field-rate": 60,
        "ndi-audio-sample-rate": 48000,
        "ndi-audio-channels": 1,
        "ndi-audio-bit-count": 16,
        "upgrade-ret": 108,
        "upgrade-step": 0,
        "upgrade-percent": 0,
        "usb-option": 0,
        "sd-option": 0,
        "CPUStatus": 15,
        "NetStatus": 87701504,
        "MemStatus": 59,
      }
    }
  ]
}
```

```
    "Net2Status": 0,  
    "Net3Status": 0  
  },  
  "is-upload-file": 0,  
  "upload-file-pos": 0,  
  "dev-warning": 0  
} ]  
}
```

get-reject-devs接口

获取云平台上被拒绝申请的设备列表。

仅管理员有权限。

针对已拒绝申请的设备，可以进行以下操作：

- [删除设备](#)

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-reject-devs
数据格式	响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-reject-devs HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
reject-devs	审核被拒绝的设备列表。

reject-devs

参数	描述
catalog-id	设备分组ID。
sn	序列号。
module	产品族编号。
dev-warning	取值大于0，表示异常设备。 dev-warning取值与异常设备状态掩码中的所有掩码做逻辑与计算，可能有多个异常。 例如：(dev-warning & 0x01)=== 0x01，表示此设备无信号。
is-everyone	是否所有用户可见。0：否，1：是。
users	is-everyone取值为0时有效，可见用户集合。
apply-date	申请时间。
online-date	设备在线时间。大于0：设备在线，小于等于0：设备离线。
is-upload-file	是否在上传固件。1：正在上传固件到设备上。 使用参见： 设备升级流程 。
upload-file-pos	固件上传进度百分比。
status	设备的基本信息。

status

参数	描述
name	设备名称。
product-id	设备产品编号。
module-name	设备产品型号。
hardware-ver	硬件版本。
firmware-ver	固件版本。
up-time	设备上电时间，单位：秒。
cur-status	设备状态掩码，不同设备不同处理。
eth	以太网IP。
wifi	WI-FI IP。
rndis	USB RNDIS IP。
upgrade-ret	设备安装固件状态码。参见： API状态码 。 使用参见： 设备升级流程 。
upgrade-step	当前升级步骤，一共有4步。
upgrade-percent	当前设备升级进度。

响应示例


```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "result": 0,
  "reject-devs": [
    {
      "catalog-id": 0,
      "sn": "A305200908004",
      "module": 2,
      "dev-status": 1,
      "network": "",
      "is-upgrade": 0,
      "users": null,
      "is-everyone": 1,
      "apply-date": 1635152956,
      "online-date": 1635211116804,
      "offline-date": 1635165703,
      "cpu": [],
      "mem": [],
      "net": [],
      "net2": [],
      "net3": [],
      "sid": "681042eca9234d40455259395c1b608e",
      "is-upload-file": 0,
      "upload-file-pos": 0,
      "dev-warning": 0,
      "status": {
        "name": "Ultra Encode A305200908004",
        "product-id": 773,
        "module-name": "Ultra Encode SDI",
        "hardware-ver": "A",
        "firmware-ver": "1.3.484",
        "conn-ip": "",
        "up-time": 85854,
        "eth": "10.10.8.36",
        "sd-total": 0,
        "cur-status": 64,
        "wifi": "",
        "input": "1920x1080p, 60.00 Hz",
        "output": "",
        "usb-total": 0,
        "usb-free": 0,
        "usb-used": 0,
        "usb-status": 0,
        "usb-usage": 0,
        "sd-free": 0,
        "sd-used": 0,
        "sd-status": 0,
        "sd-usage": 0,
        "rec-codec": "1920x1080, By Input, 16384 Kbps",
        "rec-sec": 0,
        "rec-kbps": 0,
        "live-codec": "1280x720, 30.00, 1024 Kbps",
        "live-sec": 0,
        "live-kbps": 0,
        "ndi-connected": false,
        "ndi-url": "",
        "ndi-enabled": false,
        "rndis": "",
        "ndi-name": "",
        "ndi-video-width": 0,
        "ndi-video-height": 0,
        "ndi-video-scan": "",
        "ndi-video-field-rate": 0,
        "ndi-audio-sample-rate": 0,
        "ndi-audio-channels": 0,
        "ndi-audio-bit-count": 0,
        "upgrade-ret": 27,
        "upgrade-step": 0,
        "upgrade-percent": 0,
        "usb-option": 0,
        "sd-option": 0,
      }
    }
  ]
}
```

```
"CPUStatus": 23,  
"NetStatus": 279440,  
"MemStatus": 37,  
"Net2Status": 0,  
"Net3Status": 0  
}  
}  
]  
}
```

accept-dev接口

同意待审核设备申请。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/registry/accept-dev
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
sn	指定审核通过的设备序列号集合。

请求示例

```
POST /sc_cloud/registry/accept-dev HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json; charset=utf-8
Content-Length: 24
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "sn":["B401190213050"]
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: accept-dev。

响应示例

```
HTTP/1.1 200 OK
content-type: application/json; charset=utf-8
content-length: 34
{
  "method": "accept-dev",
  "result": 0
}
```

reject-dev接口

拒绝待审核的设备。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/registry/reject-dev
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
sn	指定拒绝的设备序列号集合。

请求示例

```
POST /sc_cloud/registry/reject-dev HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json; charset=utf-8
Content-Length: 24
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "sn":["A305200908004"]
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: reject-dev。

响应示例

```
HTTP/1.1 200 OK
content-type: application/json; charset=utf-8
content-length: 34
{
  "method": "reject-dev",
  "result": 0
}
```

delete-dev接口

云平台删除设备，设备包含未审核的设备、审核通过的设备、拒绝的设备等。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/registry/delete-dev
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
sn	指定删除的设备序列号集合。

请求示例

```
POST /sc_cloud/registry/delete-dev HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json; charset=utf-8
Content-Length: 24
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "sn":["A304200731001"]
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: delete-dev。

响应示例

```
HTTP/1.1 200 OK
content-type: application/json; charset=utf-8
content-length: 34
{
  "method": "delete-dev",
  "result": 0
}
```


get-devs接口

获取通过审核的设备列表。

通过审核后，设备可以通过云平台远程管理：

- [启动/停止](#)录制。
- [启动/停止](#)直播。
- [移动到](#)指定的设备分组。
- 对设备进行[升级](#)、[重启](#)、[删除](#)操作。
- 调出设备的[WebGUI工具](#)，进行详细的参数配置。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-devs?id=0&module=0
数据格式	响应消息：Content-Type: application/json

请求URL参数

参数	描述
id	设备分组ID 取值为0时，表示所有审核通过的设备。
module	产品族编号 。

id 和 module 同时取值为0，表示查询所有已通过审核的设备。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-devs?id=0&module=0 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为：'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0：成功。• 其它：参见API状态码。
devs	已待审核的设备列表。

devs

参数	描述
catalog-id	设备分组ID。
sn	序列号。
module	产品族编号。
dev-warning	取值大于0，表示异常设备。 dev-warning取值与异常设备状态掩码中的所有掩码做逻辑与计算，可能有多个异常。 例如：(dev-warning & 0x01)=== 0x01，表示此设备无信号。
is-everyone	是否所有用户可见。0：否，1：是。
users	is-everyone取值为0时有效，可见用户集合。
apply-date	申请时间。
online-date	设备在线时间。大于0：设备在线，小于等于0：设备离线。
is-upload-file	是否在上传固件。1：正在上传固件到设备上。 使用参见： 设备升级流程 。
upload-file-pos	固件上传进度百分比。
status	设备的基本信息。

status

参数	描述
name	设备名称。
product-id	设备产品编号。
module-name	设备产品型号。
hardware-ver	硬件版本。
firmware-ver	固件版本。
up-time	设备上电时间，单位：秒。
cur-status	设备状态掩码，不同设备不同处理。
eth	以太网IP。
wifi	WI-FI IP。
rndis	USB RNDIS IP。
upgrade-ret	设备安装固件状态码。取值参见： API状态码 。 使用参见： 设备升级流程 。
upgrade-step	当前升级步骤，一共有4步。
upgrade-percent	当前设备升级进度。

设备升级流程

1. 上传云平台中的对应固件到设备上；
2. 上传成功后，设备自动安装和自动重启。

[module](#)为Pro Convert编码器、Pro Convert解码器时，处理流程如下：

```

getConvertUpgradeStatus (ret, step, percent) {
// ret: upgrade-ret step: upgrade-step percent:upgrade-percent
let desc = ''
let progress = 0
switch (ret) {
case 106: // 正在升级...
    desc = 'upgrading'
    progress = percent
    break
case 107: // 升级失败
    desc = 'upgrade-err'
    break
case 109:
    desc = 'rebooting'
    progress = 100
    break
// case 108: // 初始化升级状态
// stepTitle = ''
// progress = 0
// break
}
return { step, desc, progress, isUpdating: desc !== '' }
}

```

module为Ultra Stream、Ultra Encode时，处理流程如下：

```

getUpgradeStatus (ret, step, percent) {
// ret: upgrade-ret step: upgrade-step percent:upgrade-percent
let desc = ''
let progress = 0
switch (ret) {
case 2:
    switch (step) {
        case 1: // download
            desc = 'upgrading'
            progress = Math.ceil(percent * 2 / 10)
            break
        case 2: // install
            desc = 'upgrading'
            progress = 20 + Math.ceil(percent * 0.78)
            break
        case 3: // install
            desc = 'upgrading'
            progress = 98 + Math.ceil(percent * 0.02)
            break
        case 4: // rebooting
            desc = 'rebooting'
            progress = 100
    }
    break
case 0:
    if (step === 4) {
        desc = 'rebooting'
        progress = 100
    }
    break
// case 5:
// case 27: // 初始化升级状态
// stepTitle = ''
// progress = 0
// break
}
return { step, desc, progress, isUpdating: desc !== '' }
}

```

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "result": 0,
  "devs": [
    {
      "catalog-id": 0,
      "sn": "C301191111021",
      "module": 1,
      "dev-status": 2,
      "network": "",
      "is-upgrade": 0,
      "users": null,
      "is-everyone": 1,
      "apply-date": 1634797918,
      "online-date": 1635294552,
      "offline-date": 0,
      "is-upload-file": 0,
      "upload-file-pos": 0,
      "dev-warning": 0,
      "sid": "7e09c8036bea4a10d7e428a3403533e7",
      "cpu": [],
      "mem": [],
      "net": [],
      "net2": [],
      "net3": [],
      "status": {
        "name": "Ultra Stream C301191111021s",
        "product-id": 769,
        "module-name": "Ultra Stream HDMI",
        "hardware-ver": "D",
        "firmware-ver": "1.5.170",
        "conn-ip": "",
        "up-time": 5472,
        "eth": "192.168.123.47",
        "sd-total": 31895175168,
        "cur-status": 6,
        "wifi": "",
        "input": "1920x1080i, 59.95 Hz",
        "output": "",
        "usb-total": 0,
        "usb-free": 0,
        "usb-used": 0,
        "usb-status": 0,
        "usb-usage": 0,
        "sd-free": 30733156352,
        "sd-used": 88276992,
        "sd-status": 1,
        "sd-usage": 0,
        "rec-codec": "1920x1080, 29.97, 2048 Kbps",
        "rec-sec": 13448,
        "rec-kbps": 0,
        "live-codec": "1280x720, 30.00, 1024 Kbps",
        "live-sec": 23192,
        "live-kbps": 138703,
        "ndi-connected": false,
        "ndi-url": "",
        "ndi-enabled": false,
        "rndis": "",
        "ndi-name": "",
        "ndi-video-width": 0,
        "ndi-video-height": 0,
        "ndi-video-scan": "",
        "ndi-video-field-rate": 0,
        "ndi-audio-sample-rate": 0,
        "ndi-audio-channels": 0,
        "ndi-audio-bit-count": 0,
        "upgrade-ret": 27,
        "upgrade-step": 0,
        "upgrade-percent": 0,
        "usb-option": 0,
        "sd-option": 1,
      }
    }
  ]
}
```

```
"CPUStatus": 35,  
"NetStatus": 1259264,  
"MemStatus": 48,  
"Net2Status": 0,  
"Net3Status": 0  
}  
}  
]  
}
```

get-warning-devs接口

获取异常设备列表。

异常设备有：

- 离线设备。
- 其他，参见[异常设备状态掩码](#)。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-warning-devs?module=0
数据格式	响应消息：Content-Type: application/json

请求URL参数

参数	描述
module	产品族编号 。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-warning-devs?module=0 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为：'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0：成功。• 其它：参见API状态码。
warning-devs	异常设备列表。

warning-devs

参数	描述
catalog-id	设备分组ID。
sn	序列号。
module	产品族编号。
dev-warning	取值大于0，表示异常设备。 dev-warning取值与异常设备状态掩码中的所有掩码做逻辑与计算，可能有多个异常。 例如：(dev-warning & 0x01)=== 0x01，表示此设备无信号。
is-everyone	是否所有用户可见。0：否，1：是。
users	is-everyone取值为0时有效，可见用户集合。
apply-date	申请时间。
online-date	设备在线时间。大于0：设备在线，小于等于0：设备离线。
is-upload-file	是否在上传固件。1：正在上传固件到设备上。 使用参见： 设备升级流程 。
upload-file-pos	固件上传进度百分比。
status	设备的基本信息。

status

参数	描述
name	设备名称。
product-id	设备产品编号。
module-name	设备产品型号。
hardware-ver	硬件版本。
firmware-ver	固件版本。
up-time	设备上电时间，单位：秒。
cur-status	设备状态掩码，不同设备不同处理。
eth	以太网IP。
wifi	WI-FI IP。
rndis	USB RNDIS IP。
upgrade-ret	设备安装固件状态码。取值参见： API状态码 。 使用参见： 设备升级流程 。
upgrade-step	当前升级步骤，一共有4步。
upgrade-percent	当前设备升级进度。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "result": 0,
  "warning-devs": [
    {
      "catalog-id": 2,
      "sn": "D301210104248",
      "module": 1,
      "dev-status": 0,
      "network": "",
      "is-upgrade": 0,
      "users": null,
      "is-everyone": 0,
      "apply-date": 1635216013,
      "online-date": 1635402331798,
      "offline-date": 1635402327768,
      "cpu": [],
      "mem": [],
      "net": [],
      "net2": [],
      "net3": [],
      "sid": "2ee065cb8edd6c3e63d3fa72e5634acd",
      "status": {
        "name": "Ultra Stream D301210104248",
        "product-id": 769,
        "module-name": "Ultra Stream HDMI",
        "hardware-ver": "D",
        "firmware-ver": "1.5.179",
        "conn-ip": "",
        "up-time": 103558,
        "eth": "10.10.10.122",
        "sd-total": 31249055744,
        "cur-status": 64,
        "wifi": "192.168.48.1",
        "input": "No Signal",
        "output": "",
        "usb-total": 0,
        "usb-free": 0,
        "usb-used": 0,
        "usb-status": 0,
        "usb-usage": 0,
        "sd-free": 9576153088,
        "sd-used": 20599160832,
        "sd-status": 1,
        "sd-usage": 68,
        "rec-codec": "1920x1080, 30.00, 2048 Kbps",
        "rec-sec": 0,
        "rec-kbps": 0,
        "live-codec": "1280x720, 30.00, 1024 Kbps",
        "live-sec": 0,
        "live-kbps": 0,
        "ndi-connected": false,
        "ndi-url": "",
        "ndi-enabled": false,
        "rndis": "",
        "ndi-name": "",
        "ndi-video-width": 0,
        "ndi-video-height": 0,
        "ndi-video-scan": "",
        "ndi-video-field-rate": 0,
        "ndi-audio-sample-rate": 0,
        "ndi-audio-channels": 0,
        "ndi-audio-bit-count": 0,
        "upgrade-ret": 27,
        "upgrade-step": 0,
        "upgrade-percent": 0,
        "usb-option": 1,
        "sd-option": 1,
        "CPUStatus": 26,
        "NetStatus": 72296,
        "MemStatus": 10,
      }
    }
  ]
}
```



```
    "Net2Status": 10328,  
    "Net3Status": 0  
  },  
  "is-upload-file": 0,  
  "upload-file-pos": 0,  
  "dev-warning": 3  
} ]  
}
```

设备重启

重启指定设备。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/path?method=reboot&sc-client-id=A302190930002
数据格式	响应消息: Content-Type: application/json

- 不同产品族请求URL的path定义:

产品族	值
Ultra Stream	socket/usapi
Ultra Encode	socket/usapi
Pro Convert编码器	socket/mwapi
Pro Convert解码器	socket/mwapi

- 请求URL参数说明

参数	描述
method	方法名: reboot。
sc-client-id	指定重启的设备序列号。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

不同设备的重启方式不一样。

- Ultra Stream、Ultra Encode:

```
GET /socket/usapi?method=reboot&sc-client-id=A302190930002 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin
```

- Pro Convert编码器、Pro Convert解码器:

```
GET /socket/mwapi?method=reboot&sc-client-id=B401190213050 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none"> • 0: 成功。 • 1: 重复操作。 • 其它: 参见API状态码。
status	返回码。 <ul style="list-style-type: none"> • 0: 成功。 • 1: 重复操作。 • 其它: 参见API状态码。

响应示例

不同设备的响应不一样。

- Ultra Stream、Ultra Encode:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "result": 0
}
```

- Pro Convert编码器、Pro Convert解码器:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "status": 0
}
```

设备重命名

重命名指定设备。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/path?method=xxx&name=xxx&sc-client-id=xxx
数据格式	响应消息: Content-Type: application/json

请求URL参数

- 不同产品族对应的URL参数不同。

产品族	path	method
Ultra Stream	socket/usapi	set-name
Ultra Encode	socket/usapi	set-name
Pro Convert编码器	socket/mwapi	set-eth-config
Pro Convert解码器	socket/mwapi	set-eth-config

- 参数说明如下:

参数	描述
method	方法名。参见上表method。
name	设备的新名称。
sc-client-id	重命名设备的序列号。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

不同设备修改名称的方式不一样。

- Ultra Stream、Ultra Encode:

```
GET /socket/usapi?method=set-name&name=Encode&sc-client-id=A305200908004 HTTP/1.1
Accept: application/json, text/plain, */*
Connection: keep-alive
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin
```

- Pro Convert编码器、Pro Convert解码器:

```
GET /socket/mwapi?method=set-eth-config&name=Convert&sc-client-id=A421210928002 HTTP/1.1
Accept: application/json, text/plain, */*
Connection: keep-alive
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
status	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。

响应示例

不同设备的响应不一样。

- Ultra Stream、Ultra Encode:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "result": 0
}
```

- Pro Convert编码器、Pro Convert解码器:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "status": 0,
  "reconnect": true
}
```

start-upgrade

升级指定设备。本接口支持批量升级。

批量升级要求设备的[产品族编号](#)、[产品编号](#)和硬件版本都相同。

仅管理员有权限。

升级前必须确定云平台上指定设备的固件版本。可通过接口[get-all-firmwares](#)查询到指定设备的所有固件版本。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/upgrade/start
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
sn	指定升级的设备序列号集合。
firmware-id	固件ID 。通过接口 get-all-firmwares 获得。

请求示例

```
POST /sc_cloud/upgrade/start HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 41
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "sn":["A302190930002"],
  "firmware-id":21
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: start-upgrade。
details	设备升级结果集。

details

设备升级结果集。

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 1: 重复操作。• 其它: 参见API状态码。
sn	升级设备的序列号。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 83
{
  "result": 0,
  "method": "start-upgrade",
  "details": [
    {
      "sn": "A302190930002",
      "result": 0
    }
  ]
}
```

start-rec接口

开启录制功能。

仅管理员有权限。

开启后可以调用[stop-rec](#)接口进行停止。

本接口适用以下产品族：

- Ultra Stream

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/socket/usapi?method=start-rec&sc-client-id=xxx
数据格式	响应消息：Content-Type: application/json

请求URL参数

参数	描述
method	方法名：start-rec。
sc-client-id	指定开启录制的设备序列号。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /socket/usapi?method=start-rec&sc-client-id=D301210104248 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin
```

响应参数

Response Header

参数	描述
Content-Type	取值为：'application/json'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0：成功。• 1：重复操作。• 其它：参见API状态码。

响应示例


```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "result": 0
}
```

stop-rec接口

停止录制功能。

仅管理员有权限。

本接口适用以下产品族：

- Ultra Stream

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/socket/usapi?method=stop-rec&sc-client-id=xxx
数据格式	响应消息：Content-Type: application/json

请求URL参数

参数	描述
method	方法名：stop-rec。
sc-client-id	指定停止录制的设备序列号。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /socket/usapi?method=stop-rec&sc-client-id=D301210104248 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin
```

响应参数

Response Header

参数	描述
Content-Type	取值为：'application/json'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0：成功。• 1：重复操作。• 其它：参见API状态码。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "result": 0
}
```

start-live接口

开启直播功能。

仅管理员有权限。

开启后可以调用[stop-live](#)接口进行停止直播。

本接口适用以下产品族：

- Ultra Stream

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/socket/usapi?method=start-live&sc-client-id=xxx
数据格式	响应消息: Content-Type: application/json

请求URL参数

参数	描述
method	方法名: start-live。
sc-client-id	指定开启直播的设备序列号。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /socket/usapi?method=start-live&sc-client-id=D301210104248 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 1: 重复操作。• -53: 直播配置错误。• 其它: 参见API状态码。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "result": 0
}
```

stop-live接口

停止直播功能。

仅管理员有权限。

本接口适用以下产品族：

- Ultra Stream

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/socket/usapi?method=stop-live&sc-client-id=xxx
数据格式	响应消息：Content-Type: application/json

请求URL参数

参数	描述
method	方法名：stop-live。
sc-client-id	指定停止直播的设备序列号。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /socket/usapi?method=stop-live&sc-client-id=D301210104248 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin
```

响应参数

Response Header

参数	描述
Content-Type	取值为：'application/json'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0：成功。• 1：重复操作。• 其它：参见API状态码。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "result": 0
}
```

get-dev接口

获取指定设备的详细信息，此设备必须是通过云平台审核的。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-dev?sn=xxx
数据格式	响应消息: Content-Type: application/json

请求URL参数

参数	描述
sn	设备序列号。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-dev?sn=A302190930002 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
dev	指定设备的详细信息。

dev

参数	描述
catalog-id	设备分组ID。
sn	序列号。
module	产品族编号。
dev-warning	取值大于0，表示异常设备。 dev-warning取值与异常设备状态掩码中的所有掩码做逻辑与计算，可能有多个异常。 例如：(dev-warning & 0x01)=== 0x01，表示此设备无信号。
is-everyone	是否所有用户可见。0：否，1：是。
users	is-everyone取值为0时有效，可见用户集合。
apply-date	申请时间。
online-date	设备在线时间。大于0：设备在线，小于等于0：设备离线。
is-upload-file	是否在上传固件。1：正在上传固件到设备上。 使用参见： 设备升级流程 。
upload-file-pos	固件上传进度百分比。
status	设备的基本信息。

status

参数	描述
name	设备名称。
product-id	设备产品编号。
module-name	设备产品型号。
hardware-ver	硬件版本。
firmware-ver	固件版本。
up-time	设备上电时间，单位：秒。
cur-status	设备状态掩码，不同设备不同处理。
eth	以太网IP。
wifi	WI-FI IP。
rndis	USB RNDIS IP。
upgrade-ret	设备安装固件状态码。取值参见： API状态码 。 使用参见： 设备升级流程 。
upgrade-step	当前升级步骤，一共有4步。
upgrade-percent	当前设备升级进度。
input	输入信号格式，包括分辨率、帧率等。
output	输出信号格式，包括分辨率、帧率等。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 1600
Connection: keep-alive
{
  "result": 0,
  "dev": {
    "catalog-id": 2,
    "sn": "A302190930002",
    "module": 1,
    "dev-status": 2,
    "network": "",
    "is-upgrade": 0,
    "users": [
      "Admin"
    ],
    "is-everyone": 0,
    "apply-date": 1634088523,
    "online-date": 1635413950,
    "offline-date": 0,
    "is-upload-file": 0,
    "upload-file-pos": 100,
    "dev-warning": 1,
    "sid": "245afe8cc4362ad0355f672433ff8608",
    "cpu": [],
    "mem": [],
    "net": [],
    "net2": [],
    "net3": [],
    "status": {
      "name": "Ultra Stream A302190930002",
      "product-id": 770,
      "module-name": "Ultra Stream SDI",
      "hardware-ver": "A",
      "firmware-ver": "1.5.178",
      "conn-ip": "",
      "up-time": 104370,
      "eth": "",
      "sd-total": 32217448448,
      "cur-status": 0,
      "wifi": "10.10.3.34",
      "input": "No Signal",
      "output": "",
      "usb-total": 0,
      "usb-free": 0,
      "usb-used": 0,
      "usb-status": 0,
      "usb-usage": 0,
      "sd-free": 18495471616,
      "sd-used": 12648235008,
      "sd-status": 1,
      "sd-usage": 40,
      "rec-codec": "1280x720, 59.94, 2048 Kbps",
      "rec-sec": 0,
      "rec-kbps": 0,
      "live-codec": "1280x720, 30.00, 1024 Kbps",
      "live-sec": 0,
      "live-kbps": 0,
      "ndi-connected": false,
      "ndi-url": "",
      "ndi-enabled": false,
      "rndis": "",
      "ndi-name": "",
      "ndi-video-width": 0,
      "ndi-video-height": 0,
      "ndi-video-scan": "",
      "ndi-video-field-rate": 0,
      "ndi-audio-sample-rate": 0,
      "ndi-audio-channels": 0,
      "ndi-audio-bit-count": 0,
      "upgrade-ret": 27,
      "upgrade-step": 0,
    }
  }
}
```

```
"upgrade-percent": 0,  
"usb-option": 0,  
"sd-option": 1,  
"CPUStatus": 18,  
"NetStatus": 0,  
"MemStatus": 44,  
"Net2Status": 2328,  
"Net3Status": 0  
}  
}  
}
```

set-dev-secu接口

指定设备的安全管理接口。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/registry/set-dev-secu
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
sn	设备序列号数组。
param-int	设备是否所有用户可见。0: 否, 1: 是。
params	param-int取值为0时有效, 可见用户集合。

params

- param-int取值为1时, params取值为: [""]。
- param-int取值为0时, params为指定用户名的数组。

请求示例

```
POST /sc_cloud/registry/set-dev-secu HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 64
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "sn": [
    "A302190930002"
  ],
  "param-int": 0,
  // 指定设备只有用户名为"Admin"和"Test"的用户可见
  "params": [
    "Admin",
    "Test"
  ]
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
method	方法名: set-dev-secu。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 36
Connection: keep-alive
{
  "method": "set-dev-secu",
  "result": 0
}
```

get-dev-logs接口

获取指定设备的运行日志。

如需清除设备日志，可以调用[clear-logs](#)接口。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-dev-logs?sn=xxx
数据格式	响应消息: Content-Type: application/json

请求URL参数

参数	描述
sn	设备序列号。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-dev-logs?sn=D301210104248 HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
logs	设备的日志列表。

logs

参数	描述
id	日志ID。
date	日志产生时间。
info	日志信息。
type	日志类型。1: 信息, 2: 警告, 3: 错误。
sn	设备序列号。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "result": 0,
  "logs": [
    {
      "id": 3630,
      "date": 1635217459,
      "info": "The device is offline.",
      "type": 2,
      "sn": "D301210104248"
    },
    {
      "id": 3631,
      "date": 1635218146,
      "info": "The device is online.",
      "type": 1,
      "sn": "D301210104248"
    }
  ]
}
```

WebGUI工具

设备在线的前提下，可以使用设备的WebGUI工具，进行详细的参数配置。

接口定义

协议	HTTP/HTTPS GET
URL	ip[:port]/socket/sc-remote/<设备序列号>/

调用实例

第一步 登录云平台，获取到Cookie；

第二步 用链接的方式打开WebGUI工具，实例代码如下：

```
<a href="http://ip[:port]/socket/sc-remote/D301210104248/" target="_blank">  
  WebGUI工具  
</a>
```


get-srt-server接口

获取SRT服务器列表。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-srt-server
数据格式	响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-srt-server HTTP/1.1
Accept: application/json, text/plain, */*
Connection: keep-alive
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
status	SRT服务器状态。

status

参数	描述
is-running	SRT服务器是否开启。 开启SRT服务器 。 关闭SRT服务器 。
connections	启动的SRT服务器数量。
down-bitrate	总播放比特率。
up-bitrate	总上传比特率。
clients	SRT服务器列表详细信息。

clients

参数	描述
name	流名称。
player	播放服务器列表。
publisher	上传服务器详情。

player

参数	描述
addr	播放服务器地址。
bitrate	播放比特率。
port	端口。
stream_name	流名称。
type	常量值：player。
uptime	运行时间，单位：毫秒。

publisher

参数	描述
addr	上传服务器地址。
bitrate	上传比特率。
port	端口。
stream_name	流名称。
type	常量值：publisher。
uptime	运行时间，单位：毫秒。

响应示例

本例中包含两个播放端。

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 487
{
  "result": 0,
  "status": {
    "is-running": true,
    "connections": 3,
    "down-bitrate": 1228800,
    "up-bitrate": 609280,
    "clients": [
      {
        "name": "streamID",
        "player": [
          {
            "addr": "10.10.14.135",
            "bitrate": 614400,
            "port": 46660,
            "stream_name": "streamID",
            "type": "player",
            "uptime": 13600
          },
          {
            "addr": "10.10.15.84",
            "bitrate": 614400,
            "port": 34154,
            "stream_name": "streamID",
            "type": "player",
            "uptime": 13602
          }
        ]
      },
      {
        "name": "publisher",
        "publisher": {
          "addr": "10.10.6.142",
          "bitrate": 609280,
          "port": 40089,
          "stream_name": "streamID",
          "type": "publisher",
          "uptime": 13616
        }
      }
    ]
  }
}
```

set-srt-settings接口

配置SRT服务器。

仅管理员有权限。

SRT服务器使用注意事项

- 必须使用SRT Caller模式推流和取流。
- 推流端和取流端的端口号、流名称、加密方式和密码必须一致。

流ID格式:

推流端流ID: <上传域名称>/live/<流名称>

取流端流ID: <播放域名称>live/<流名称>

- 推流端和取流端必须能ping通云平台，且可为任何支持SRT Caller功能的软硬件产品。

推流和取流设备均无需注册到云平台。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/srtserver/set-settings
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
port	端口, 范围: 1024~65535。
domain_player	播放域名称。
domain_publisher	上传域名称。
pbkeylen	是否使用加密, 默认不使用。取值如下: <ul style="list-style-type: none"> • 0: 不使用 • 16: AES-128 • 24: AES-192 • 32: AES-256
passphrase	加密密码。pbkeylen取值不为0时, 需要输入10~79位字符的密码。
app_player_name	播放应用名称。 <ul style="list-style-type: none"> • get-srt-server-settings接口返回。 • 不支持修改此项。
app_publisher_name	上传应用名称。 <ul style="list-style-type: none"> • get-srt-server-settings接口返回。 • 不支持修改此项。
idle_streams_timeout	超时时间。 <ul style="list-style-type: none"> • get-srt-server-settings接口返回。 • 不支持修改此项。
latency	延迟时间 <ul style="list-style-type: none"> • get-srt-server-settings接口返回。 • 不支持修改此项。
max_connections	最大连接数。 <ul style="list-style-type: none"> • get-srt-server-settings接口返回。 • 不支持修改此项。

上传流ID: <domain_publisher>/<app_publisher_name>/<流名称>
播放流ID: <domain_player>/<app_player_name>/<流名称>

请求示例

```
POST /sc_cloud/srtserver/set-settings HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 227
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "app_player_name": "live",
  "app_publisher_name": "live",
  "domain_player": "play.srt",
  "domain_publisher": "upload.srt",
  "idle_streams_timeout": 10,
  "latency": 20,
  "port": 8001,
  "max_connections": 20,
  "pbkeylen": 16,
  "passphrase": "111111111111"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
method	方法名: set-srt-settings。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 40
Connection: keep-alive
{
  "method": "set-srt-settings",
  "result": 0
}
```

get-srt-server-settings接口

获取SRT服务器的设置信息。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-srt-server-settings
数据格式	响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-srt-server-settings HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
settings	SRT服务器设置。

settings

参数	描述
app_player_name	常量值: live。
app_publisher_name	常量值: live。
domain_player	播放域名称。
domain_publisher	上传域名称。
latency	延迟时间。取值范围是20ms~8000ms, 默认为120ms。
port	端口, 范围: 1024~65535。
pbkeylen	是否使用加密, 默认不使用。取值如下: <ul style="list-style-type: none"> • 0: 不使用 • 16: AES-128 • 24: AES-192 • 32: AES-256
passphrase	加密密码。pbkeylen取值不为0时, 需要输入10~79位字符的密码。
max_connections	最大连接数。

上传流ID: <domain_publisher>/<app_publisher_name>/<流名称>
播放流ID: <domain_player>/<app_player_name>/<流名称>

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 238
{
  "result": 0,
  "settings": {
    "app_player_name": "live",
    "app_publisher_name": "live",
    "domain_player": "play.srt",
    "domain_publisher": "upload.srt",
    "idle_streams_timeout": 10,
    "latency": 20,
    "port": 8000,
    "pbkeylen": 0,
    "passphrase": "",
    "max_connections": 20
  }
}
```


start-srt-server接口

开启SRT服务器。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/srtserver/start
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

空

请求示例

```
POST /sc_cloud/srtserver/start HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 0
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: start-srt-server。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 42
Connection: keep-alive
{
  "method": "start-srt-server",
  "result": 0
}
```


stop-srt-server接口

关闭SRT服务，此时所有SRT连接都将中断。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/srtserver/stop
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

空

请求示例

```
POST /sc_cloud/srtserver/stop HTTP/1.1
Accept: application/json, text/plain, */*
Connection: keep-alive
Content-Length: 0
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: stop-srt-server。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 39
Connection: keep-alive
{
  "method": "stop-srt-server",
  "result": 0
}
```


get-channels接口

获取协议流转换列表。如查询请求中指定协议流ID，则返回指定协议流的详细信息。

协议流包含输入视频流和输出视频流。

输入视频流支持类型：

- 1) RTMP/RTMPS
- 2) SRT Listener

输出视频流支持类型：

- 1) RTMP/RTMPS
- 2) SRT Caller
- 3) SRT Listener

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-channels[?id=xx]
数据格式	响应消息：Content-Type: application/json

请求URL参数

参数	是否必选	描述
id	否	协议流ID。 <ul style="list-style-type: none">• 携带ID：获取指定协议流的详细信息。• 不携带ID：获取协议流转换列表。

下面以请求URL不携带ID举例。

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-channels HTTP/1.1
Accept: application/json, text/plain, */*
Connection: keep-alive
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为：'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none"> 0: 成功。 其它: 参见API状态码。
channels	协议流转换列表。

channels

参数	描述
id	协议流ID。
is-enable	协议流是否启动。
name	协议流名称。字符限制如下： <ul style="list-style-type: none"> 1~32个字符。 仅支持大小写字母、数字、特殊符号_-'[]()和空格。 不能以空格开头或结束。
source	输入视频流详情。
sink	输出视频流详情。

source

参数	描述
id	输入视频流ID。
is-enable	输入视频流是否启动。
name	输入视频流名称。暂不支持修改，默认值：def-channel-src-name。
type	协议流类型。取值如下： <ul style="list-style-type: none"> 1: RTMP/RTMPS 2: SRT Caller 3: SRT Listener
status	连接状态。
uptime	运行时间，单位为ms。
bitrate	传输比特率，单位为bps。
rtmp	type 取值为1时，此项有效，表示输入视频流为RTMP/RTMPS类型。
srt	type 取值为3时，此项有效，表示输入视频流为SRT Listener类型。

sink

sink是一个Object类型。需要对对象做属性遍历，才能获取到sink下的输出视频流列表。

- Object的每一个key都是一个输出视频流ID。
- Object的每一个key对应的值都是一个输出视频流的详细信息。参数说明如下：

参数	描述
id	输出视频流ID。
is-enable	输出视频流是否启动。
name	输出视频流名称。字符限制如下： <ul style="list-style-type: none"> • 1~32个字符。 • 仅支持大小写字母、数字、特殊符号_-'[]()和空格。 • 不能以空格开头或结束。
type	协议流类型。取值如下： <ul style="list-style-type: none"> • 1: RTMP/RTMPS • 2: SRT Caller • 3: SRT Listener
status	连接状态。
uptime	运行时间，单位为ms。
bitrate	传输比特率，单位为bps。
rtmp	type取值为1时，此项有效，表示输出视频流为RTMP/RTMPS类型。
srt	type取值为3时，此项有效，表示输出视频流为SRT Listener类型。 type取值为2时，此项有效，表示输出视频流为SRT Caller类型。

rtmp

参数	描述
type	常量值：1。
url	服务器地址。
key	流密钥。
is-auth	认证状态。 <ul style="list-style-type: none"> • 0: 无需认证。 • 1: 需要认证，用户名和密码必填。
user	用户名。
passwd	密码。

srt

参数	描述
ip	IP地址或域名。仅SRT Caller类型，才输入Listener端地址。
port	端口号，取值范围是1~65535。
latency	延迟时间。取值范围是20ms~8000ms，默认为120ms。
bandwidth	SRT开销占总带宽的比例。取值范围是5%~100%，默认值为25%。
stream-id	流ID。 仅SRT Caller类型有此属性。 取值范围是0~256个字符。必须与发送端的Stream ID一致。
enc-type	加密类型，默认不加密。取值对应关系如下： <ul style="list-style-type: none"> • 0: 不使用加密 • 16: AES-128 • 24: AES-192 • 32: AES-256
enc-pass	加密密码。enc-type取值不为0时，需要输入10~79位字符的密码。

输入/输出视频流连接状态码

状态值	说明
0	空闲。当前没有连接。
1	正在解析域名。
2	正在连接。
3	正在握手。
4	连接数据流。
5	连接就绪。
6	状态异常。
7	域名解析失败。
8	TCP端口连接失败。
9	握手失败。
10	连接失败。
11	授权失败。
100	监听端口失败。
101	连接已断开。
102	远端拒绝连接请求。

响应示例


```

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
{
  "result": 0,
  "channels": [
    {
      "id": 6,
      "is-enable": 1,
      "name": "Stream Conversion 0001",
      "source": {
        "id": 8,
        "is-enable": 1,
        "name": "def-channel-src-name",
        "type": 3,
        "status": 5,
        "bitrate": 0,
        "uptime": 0,
        "rtmp": {
          "type": 0, "url": "", "key": "",
          "is-auth": 0, "user": "", "passwd": "", "param": ""
        },
        "srt": {
          "ip": "0.0.0.0", "port": 8666, "latency": 120, "bandwidth": 0,
          "stream-id": "", "enc-type": 0, "enc-pass": "", "Param": ""
        }
      },
      "sink": {
        "19": {
          "id": 19,
          "is-enable": 1,
          "name": "rtmp",
          "type": 1,
          "status": 0,
          "bitrate": 0,
          "uptime": 0,
          "rtmp": {
            "type": 1, "url": "rtmps://10.0.1.32/live", "key": "234",
            "is-auth": 0, "user": "", "passwd": "", "param": ""
          },
          "srt": {
            "ip": "", "port": 0, "latency": 0, "bandwidth": 0,
            "stream-id": "", "enc-type": 0, "enc-pass": "", "Param": ""
          }
        }
      }
    },
    {
      "id": 8,
      "is-enable": 1,
      "name": "Stream Conversion 0002",
      "source": {
        "id": 10,
        "is-enable": 1,
        "name": "def-channel-src-name",
        "type": 1,
        "status": 5,
        "bitrate": 0,
        "uptime": 1698,
        "rtmp": {
          "type": 1, "url": "rtmps://10.0.1.32:1936/live", "key": "22",
          "is-auth": 0, "user": "", "passwd": "", "param": ""
        },
        "srt": {
          "ip": "", "port": 0, "latency": 0, "bandwidth": 0,
          "stream-id": "", "enc-type": 0, "enc-pass": "", "Param": ""
        }
      },
      "sink": {
        "11": {
          "id": 11,
          "is-enable": 1,
          "name": "rtmp",

```

```

    "type": 1,
    "status": 0,
    "bitrate": 0,
    "uptime": 0,
    "rtmp": {
      "type": 1, "url": "rtmps://10.0.1.32:1936/live", "key": "12",
      "is-auth": 0, "user": "", "passwd": "", "param": ""
    },
    "srt": {
      "ip": "", "port": 0, "latency": 0, "bandwidth": 0,
      "stream-id": "", "enc-type": 0, "enc-pass": "", "Param": ""
    }
  },
  "16": {
    "id": 16,
    "is-enable": 0,
    "name": "SRT Caller",
    "type": 2,
    "status": 0,
    "bitrate": 0,
    "uptime": 0,
    "rtmp": {
      "type": 0, "url": "", "key": "",
      "is-auth": 0, "user": "", "passwd": "", "param": ""
    },
    "srt": {
      "ip": "10.10.11.105", "port": 6666, "latency": 120, "bandwidth": 25,
      "stream-id": "", "enc-type": 0, "enc-pass": "", "Param": ""
    }
  },
  "20": {
    "id": 20,
    "is-enable": 1,
    "name": "SRT Listener",
    "type": 3,
    "status": 0,
    "bitrate": 0,
    "uptime": 0,
    "rtmp": {
      "type": 0, "url": "", "key": "",
      "is-auth": 0, "user": "", "passwd": "", "param": ""
    },
    "srt": {
      "ip": "0.0.0.0", "port": 5236, "latency": 120, "bandwidth": 0,
      "stream-id": "", "enc-type": 0, "enc-pass": "", "Param": ""
    }
  }
}
]
}

```

add-channel接口

添加输入协议视频流，支持类型如下：

- RTMP/RTMPS
- SRT Listener

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/add
数据格式	请求消息：Content-Type: application/json 响应消息：Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
name	协议流名称。字符限制如下： <ul style="list-style-type: none">• 1~32个字符。• 仅支持大小写字母、数字、特殊符号._-+[]()和空格。• 不能以空格开头或结束。
protocol	协议流详细信息。

protocol

参数	描述
name	输入视频流名称。暂不支持修改。默认值：def-channel-src-name。
is-enable	输入视频流是否启动。默认值：1。
type	协议流类型。取值如下： <ul style="list-style-type: none">• 1：RTMP/RTMPS• 3：SRT Listener
rtmp	type=1 时，Request Body参见： RTMP/RTMPS示例 。
srt	type=3 时，Request Body参见： SRT Listener示例 。

rtmp

参数	描述
type	常量值：1。
url	服务器地址。
key	流密钥。
is-auth	认证状态。 <ul style="list-style-type: none"> 0: 无需认证。 1: 需要认证，用户名和密码必填。
user	用户名。
passwd	密码。

srt

参数	描述
ip	常量值："0.0.0.0"。
port	端口号，取值范围是1~65535。
latency	延迟时间。取值范围是20ms~8000ms，默认为120ms。
enc-type	加密类型，默认不加密。取值对应关系如下： <ul style="list-style-type: none"> 0: 不使用加密 16: AES-128 24: AES-192 32: AES-256
enc-pass	加密密码。enc-type取值不为0时，需要输入10~79位字符的密码。

请求示例

- RTMP/RTMPS实例:

```
POST /sc_cloud/channel/add HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 179
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "name": "rtmp push",
  "protocol": {
    "name": "def-channel-src-name",
    "is-enable": 1,
    "type": 1,
    "rtmp": {
      "type": 1,
      "url": "rtmp://127.0.0.1/live",
      "key": "test",
      "is-auth": 0,
      "user": "",
      "passwd": ""
    }
  }
}
```

- SRT Listener实例:

```

POST /sc_cloud/channel/add HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 179
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "name": "srt listener",
  "protocol": {
    "name": "def-channel-src-name",
    "is-enable": 1,
    "type": 3,
    "srt": {
      "ip": "0.0.0.0",
      "port": 3522,
      "latency": 120,
      "enc-type": 16,
      "enc-pass": "dddddddddddd"
    }
  }
}

```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none"> 0: 成功。 其它: 参见API状态码。
id	协议流ID。
method	方法名: add-channel。

响应示例

```

HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 43
Connection: keep-alive
{
  "id": 13,
  "method": "add-channel",
  "result": 0
}

```

set-channel-source接口

修改输入视频流。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/set-source
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
name	协议流名称。 本接口不支持修改此项，可以通过 set-channel-name 接口修改。
id	协议流ID。
protocol	协议流详细信息。

protocol

参数	描述
id	输入视频流ID。
name	输入视频流名称。暂不支持修改。默认值: def-channel-src-name。
is-enable	输入视频流是否启动。 本接口不支持修改此项，可以通过 set-channel-enable 接口开启/关闭。
type	协议流类型。取值如下： <ul style="list-style-type: none">1: RTMP/RTMPS2: SRT Caller3: SRT Listener
rtmp	type=1 时，Request Body参见： RTMP/RTMPS示例 。
srt	type=3 时，Request Body参见： SRT Listener示例 。

rtmp

参数	描述
type	常量值: 1。
url	服务器地址。
key	流密钥。
is-auth	认证状态。 <ul style="list-style-type: none">0: 无需认证。1: 需要认证，用户名和密码必填。
user	用户名。
passwd	密码。

srt

参数	描述
ip	常量值: "0.0.0.0"。
port	端口号, 取值范围是1~65535。
latency	延迟时间。取值范围是20ms~8000ms, 默认为120ms。
enc-type	加密类型, 默认不加密。取值对应关系如下: <ul style="list-style-type: none">• 0: 不使用加密• 16: AES-128• 24: AES-192• 32: AES-256
enc-pass	加密密码。enc-type取值不为0时, 需要输入10~79位字符的密码。

请求示例

- RTMP/RTMPS实例:

```
POST /sc_cloud/channel/set-source HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 208
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "name": "rtmp",
  "id": 12,
  "protocol": {
    "id": 22,
    "name": "def-channel-src-name",
    "is-enable": 1,
    "type": 1,
    "rtmp": {
      "type": 1,
      "url": "rtmp://10.0.1.32/live",
      "key": "pull-modify",
      "is-auth": 0,
      "user": "",
      "passwd": ""
    }
  }
}
```

- SRT Listener实例:

```
POST /sc_cloud/channel/set-source HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 202
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "name": "srt listener",
  "id": 13,
  "protocol": {
    "id": 23,
    "name": "def-channel-src-name",
    "is-enable": 1,
    "type": 3,
    "srt": {
      "ip": "0.0.0.0",
      "port": 3522,
      "latency": 120,
      "enc-type": 16,
      "enc-pass": "ddddddddddddffff"
    }
  }
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
method	方法名: set-channel-source。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 42
Connection: keep-alive
{
  "method": "set-channel-source",
  "result": 0
}
```


set-channel-name接口

修改输入协议视频流的名称。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/set-name
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	协议流ID。执行 add-channel 接口生成的。可以通过 get-channels 接口查询到。
name	新名称。字符限制如下： <ul style="list-style-type: none">1~32个字符。仅支持大小写字母、数字、特殊符号_-'[]()和空格。不能以空格开头或结束。

请求示例

```
POST /sc_cloud/channel/set-name HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 35
Connection: keep-alive
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "name": "new name"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: set-channel-name。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 40
Connection: keep-alive
{
  "method": "set-channel-name",
  "result": 0
}
```

set-channel-enable接口

开启/关闭输入协议流转化。

- 新增输入协议流默认是开启状态。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/set-enable
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	协议流ID。执行add-channel接口生成的。可以通过get-channels接口查询到。
is-enable	协议流是否启动。 <ul style="list-style-type: none">• 1: 开启• 0: 关闭

请求示例

```
POST /sc_cloud/channel/set-enable HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 23
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "is-enable": 0
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
method	方法名: set-channel-enable。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 42
Connection: keep-alive
{
  "method": "set-channel-enable",
  "result": 0
}
```

delete-channel接口

删除指定的协议流，即删除协议流下的输入视频流和所有输出视频流。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/delete
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	协议流ID。执行 add-channel 接口生成的。可以通过 get-channels 接口查询到。

请求示例

```
POST /sc_cloud/channel/delete HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json; charset=utf-8
Content-Length: 9
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: delete-channel。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 38
Connection: keep-alive
{
  "method": "delete-channel",
  "result": 0
}
```

add-channel-sink接口

添加输出视频流，支持类型如下：

- RTMP/RTMPS
- SRT Listener
- SRT Caller

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/add-sink
数据格式	请求消息：Content-Type: application/json 响应消息：Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	协议流ID。执行 add-channel 接口生成的。可以通过 get-channels 查询到。
protocol	协议流详细信息。

protocol

参数	描述
name	输出视频流名称。
is-enable	输出视频流是否启动。默认值：1。
type	协议流类型。取值如下： <ul style="list-style-type: none">• 1: RTMP/RTMPS• 2: SRT Caller• 3: SRT Listener
rtmp	type=1 时，Request Body参见： RTMP/RTMPS示例 。
srt	type=2 时，Request Body参见： SRT Caller示例 。 type=3 时，Request Body参见： SRT Listener示例 。

rtmp

参数	描述
type	常量值：1。
url	服务器地址。
key	流密钥。
is-auth	认证状态。 <ul style="list-style-type: none">• 0: 无需认证。• 1: 需要认证，用户名和密码必填。
user	用户名。
passwd	密码。

srt

参数	描述
ip	IP地址或域名。type为SRT Listener时，IP为常量值："0.0.0.0"。
port	端口号，取值范围是1~65535。
latency	延迟时间。取值范围是20ms~8000ms，默认为120ms。
bandwidth	SRT开销占总带宽的比例。取值范围是5%~100%，默认值为25%。 仅SRT Caller类型有此属性。
stream-id	流ID。 仅SRT Caller类型有此属性。 取值范围是0~256个字符。必须与发送端的Stream ID一致。
enc-type	加密类型，默认不加密。取值对应关系如下： <ul style="list-style-type: none">• 0: 不使用加密• 16: AES-128• 24: AES-192• 32: AES-256
enc-pass	加密密码。enc-type取值不为0时，需要输入10~79位字符的密码。

请求示例

- RTMP/RTMPS实例:

```
POST /sc_cloud/channel/add-sink HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 190
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "protocol": {
    "name": "output-rtmp",
    "is-enable": 1,
    "type": 1,
    "rtmp": {
      "type": 1,
      "url": "rtmp://10.2.3.63/live",
      "key": "",
      "is-auth": 1,
      "user": "test",
      "passwd": "12345678",
      "name": "output-rtmp"
    }
  }
}
```

- SRT Caller实例:


```

POST /sc_cloud/channel/add-sink HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 225
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "protocol": {
    "name": "output-srt caller",
    "is-enable": 1,
    "type": 2,
    "srt": {
      "ip": "10.0.2.3",
      "port": 678,
      "latency": 120,
      "enc-type": 16,
      "enc-pass": "passphrase",
      "stream-id": "stream id",
      "bandwidth": 25,
      "name": "output-srt caller"
    }
  }
}

```

- SRT Listener实例:

```

POST /sc_cloud/channel/add-sink HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 178
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "protocol": {
    "name": "output-srt listener",
    "is-enable": 1,
    "type": 3,
    "srt": {
      "ip": "0.0.0.0",
      "port": 666,
      "latency": 120,
      "enc-type": 0,
      "enc-pass": "",
      "name": "output-srt listener"
    }
  }
}

```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none"> • 0: 成功。 • 其它: 参见API状态码。
method	方法名: add-channel-sink。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 40
Connection: keep-alive
{
  "method": "add-channel-sink",
  "result": 0
}
```

set-channel-sink接口

修改输出视频流。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/set-sink
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	协议流ID。执行 add-channel 接口生成的。可以通过 get-channels 接口查询到。
protocol	协议流详细信息。

protocol

参数	描述
name	输出视频流名称。 本接口不支持修改此项，可以通过 set-channel-sink-name 接口修改。
is-enable	输出视频流是否启动。 本接口不支持修改此项，可以通过 set-channel-sink-enable 接口开启/关闭。
type	协议流类型。取值如下： <ul style="list-style-type: none">1: RTMP/RTMPS2: SRT Caller3: SRT Listener
rtmp	type=1 时，Request Body参见： RTMP/RTMPS示例 。
srt	type=2 时，Request Body参见： SRT Caller示例 。 type=3 时，Request Body参见： SRT Listener示例 。

rtmp

参数	描述
type	常量值: 1。
url	服务器地址。
key	流密钥。
is-auth	认证状态。 <ul style="list-style-type: none">0: 无需认证。1: 需要认证，用户名和密码必填。
user	用户名。
passwd	密码。

srt

参数	描述
ip	ip地址或域名。type为SRT Listener时, ip为常量值: "0.0.0.0"。
port	端口号, 取值范围是1~65535。
latency	延迟时间。取值范围是20ms~8000ms, 默认为120ms。
bandwidth	SRT开销占总带宽的比例。取值范围是5%~100%, 默认值为25%。 仅SRT Caller类型有此属性。
stream-id	流ID。 仅SRT Caller类型有此属性。 取值范围是0~256个字符。必须与发送端的Stream ID一致。
enc-type	加密类型, 默认不加密。取值对应关系如下: <ul style="list-style-type: none"> • 0: 不使用加密 • 16: AES-128 • 24: AES-192 • 32: AES-256
enc-pass	加密密码。enc-type取值不为0时, 需要输入10~79位字符的密码。

请求示例

- RTMP/RTMPS实例:

```
POST /sc_cloud/channel/set-sink HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 212
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "protocol": {
    "id": 24,
    "name": "output-rtmp-new",
    "is-enable": 1,
    "type": 1,
    "rtmp": {
      "type": 1,
      "url": "rtmp://10.2.3.63/live",
      "key": "addkey",
      "is-auth": 1,
      "user": "test",
      "passwd": "12345678",
      "name": "output-rtmp"
    }
  }
}
```

- SRT Caller实例:

```

POST /sc_cloud/channel/set-sink HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 245
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "protocol": {
    "id": 25,
    "name": "output-srt caller",
    "is-enable": 1,
    "type": 2,
    "srt": {
      "ip": "10.0.2.3",
      "port": 678,
      "latency": 120,
      "enc-type": 24,
      "enc-pass": "passphrase",
      "stream-id": "modify caller",
      "bandwidth": 25,
      "name": "output-srt caller-new"
    }
  }
}

```

- SRT Listener实例:

```

POST /sc_cloud/channel/set-sink HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 206
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "protocol": {
    "id": 26,
    "name": "output-srt listener",
    "is-enable": 1,
    "type": 3,
    "srt": {
      "ip": "0.0.0.0",
      "port": 666,
      "latency": 120,
      "enc-type": 16,
      "enc-pass": "fsfsffffffff",
      "name": "output-srt listener-new"
    }
  }
}

```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none"> • 0: 成功。 • 其它: 参见API状态码。
method	方法名: set-channel-sink。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 40
Connection: keep-alive
{
  "method": "set-channel-sink",
  "result": 0
}
```

set-channel-sink-name接口

修改输出视频流的名称。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/set-sink-name
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	协议流ID。执行 add-channel 接口生成的。可以通过 get-channels 接口查询到。
protocol	协议流详细信息。

protocol

参数	描述
id	输出视频流ID。执行 add-channel-sink 接口生成的。可以通过 get-channels 接口查询到。
name	新的输出视频流名称。字符限制如下： <ul style="list-style-type: none">1~32个字符。仅支持大小写字母、数字、特殊符号_-'[]()和空格。不能以空格开头或结束。

请求示例

```
POST /sc_cloud/channel/set-sink-name HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 61
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "protocol": {
    "id": 26,
    "name": "new output name"
  }
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
method	方法名: set-channel-sink-name。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 40
Connection: keep-alive
{
  "method": "set-channel-sink-name",
  "result": 0
}
```


set-channel-sink-enable接口

开启/关闭输出视频流转化。

- 新增输出视频流默认是开启状态。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/set-sink-enable
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	协议流ID。执行add-channel接口生成的。可以通过get-channels接口查询到。
protocol	协议流详细信息。

protocol

参数	描述
id	输出视频流ID。执行add-channel-sink接口生成的。可以通过get-channels接口查询到。
is-enable	输出视频流是否启动。 <ul style="list-style-type: none">• 1: 开启• 0: 关闭

请求示例

```
POST /sc_cloud/channel/set-sink-enable HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 44
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "protocol": {
    "id": 25,
    "is-enable": 0
  }
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
method	方法名: set-channel-sink-enable。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 47
Connection: keep-alive
{
  "method": "set-channel-sink-enable",
  "result": 0
}
```

delete-channel-sink接口

删除指定的输出视频流。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/channel/delete-sink
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	协议流ID。执行 add-channel 接口生成的。可以通过 get-channels 接口查询到。
protocol	协议流详细信息。

protocol

参数	描述
id	输出视频流ID。执行 add-channel-sink 接口生成的。可以通过 get-channels 接口查询到。

请求示例

```
POST /sc_cloud/channel/delete-sink HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 30
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 13,
  "protocol": {
    "id": 26
  }
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
method	方法名: delete-channel-sink。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 43
Connection: keep-alive
{
  "method": "delete-channel-sink",
  "result": 0
}
```

set-code接口

设置邀请码接口。

云平台开启邀请码功能后，注册设备到云平台，就需要填写正确的邀请码。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/settings/set-code
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
is-code	是否开启邀请码功能。1: 是, 0: 否。
code-value	邀请码。取值是一个4位的数字。

请求示例

```
POST /sc_cloud/settings/set-code HTTP/1.1
Content-Length: 33
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=UTF-8
Cookie: mwcloud-sid=OCXWUMGEYPGIAWOWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "is-code":0,
  "code-value":"1234"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: set-code。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 32
{
  "method": "set-code",
  "result": 0
}
```

set-web接口

云平台证书管理接口。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/settings/set-web
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
is-http	是否允许 HTTP 协议访问云平台。1: 是, 0: 否。
http-port	HTTP 端口号。
is-https	是否允许 HTTPS 协议访问云平台。1: 是, 0: 否。
https-port	HTTPS 端口号。

请求示例

```
POST /sc_cloud/settings/set-web HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 58
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "is-http": 1,
  "http-port": 80,
  "is-https": 1,
  "https-port": 443
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: set-web。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 31
{
  "method": "set-web",
  "result": 0
}
```


uploadCert接口

上传CA证书，上传文件格式为.crt。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/uploadCert
数据格式	请求消息: Content-Type: multipart/form-data; boundary=xxx 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

CA证书文件。

请求示例

```
POST /sc_cloud/uploadCert HTTP/1.1
Content-Type: multipart/form-data; boundary=-----22383205084807890342375037721
Content-Length: 1510
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;

-----22383205084807890342375037721
Content-Disposition: form-data; name="file"; filename="cloudserver.crt"
Content-Type: application/x-x509-ca-cert

-----BEGIN CERTIFICATE-----
MIIDezCCAmOgAwIBAgIUQbp2VyDbpmOm7+hI07YzHX9+fQAwDQYJKoZIhvcNAQEL
BQAwVTElMAkGA1UEBhMCQ04xCzAJBgNVBAGMAkpTMQswCQYDVQQHDAJOSjERMA8G
...
YBTqm8X2bFJgDh069AtJsPSBsniHJvODMzyhtuCIgG==
-----END CERTIFICATE-----

-----22383205084807890342375037721--
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
filename	文件名。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 31
{
  "filename": "/data/cloudserver.crt",
  "result": 0
}
```

uploadCertKey接口

上传 CA私钥，上传文件格式为.pem。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/uploadCertKey
数据格式	请求消息: Content-Type: multipart/form-data; boundary=xxx 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

CA私钥文件。

请求示例

```
POST /sc_cloud/uploadCertKey HTTP/1.1
Content-Type: multipart/form-data; boundary=-----185266796315308751253203420120
Content-Length: 1920
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;

-----185266796315308751253203420120
Content-Disposition: form-data; name="file"; filename="cloudserver.pem"
Content-Type: application/octet-stream

-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAxuYuD0+i1XNHvTRV+2HLltNUNuiA15gh+OAGwJ7yhsXLrtEg
EMCbETTvtzMEgEfh1ya3fPKywDe084VT6/RIId60tZY6RC/pDayRenQhABgQQe2l6
...
aoMgVhx+L+FFm9wgKHnZrLVRjN+Z26GLXWkkkSGaoJAAIMxPPy44Jg==
-----END RSA PRIVATE KEY-----

-----185266796315308751253203420120--
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
filename	文件名。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 31
{
  "filename": "/data/cloudserver.pem",
  "result": 0
}
```

activation接口

云平台添加授权证书接口。

仅管理员有权限。

通过接口get-status返回的license-reg字段，获取证书相关信息。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/license/activation
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
sn	证书序列号。
user-name	证书标题。

请求示例

```
POST /sc_cloud/license/activation HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 56
Connection: keep-alive
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "sn": "9C4E*****F769",
  "user-name": "magewell-ltd"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: activation。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 37
{
  "result": 0,
  "method": "activation"
}
```

deactivation接口

云平台删除已授权证书接口。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/license/deactivation
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
sn	证书序列号。通过 get-status 接口返回的license-reg字段获得。
user-name	证书标题。通过 get-status 接口返回的license-reg字段获得。

请求示例

```
POST /sc_cloud/license/deactivation HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 56
Connection: keep-alive
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "sn": "9C4E*****F769",
  "user-name": "magewell-ltd"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: deactivation。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 37
{
  "result": 0,
  "method": "deactivation"
}
```


uploadFirmware接口

上传设备固件到云平台上, 上传文件格式为.mwf。

仅管理员有权限。

上传固件成功后, 需要再执行[add-firmware](#)接口使固件生效, 固件生效后才能被审核通过的同类型设备使用。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/uploadFirmware
数据格式	请求消息: Content-Type: multipart/form-data; boundary=xxx 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

设备固件文件。

请求示例

```
POST /sc_cloud/uploadFirmware HTTP/1.1
Content-Type: multipart/form-data; boundary=-----42288931224667793852045763104
Content-Length: 14556621
Cookie: mwcloud-sid=OCXWUMGEYPGIAWQOETYPNMVHDZIAZJP; mwcloud-uid=Admin;

-----42288931224667793852045763104
Content-Disposition: form-data; name="file"; filename="ultra_stream_sdi_rev_a_1_5_79.mwf"
Content-Type: application/octet-stream
...
-----42288931224667793852045763104--
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
file-path	文件路径。
file-size	文件大小。
firmware-info	上传的设备固件基本信息。

firmware-info

参数	描述
product-id	产品编号。
firmware-id	固件版本类型。 <ul style="list-style-type: none"> 取值为1，测试版本。 其他为发布版本。
marjor-ver	主版本号。
minor-ver	次版本号。
build-number	构建号。
build-time	构建时间。
product-name	产品型号。
hardware-ver	硬件版本。

固件版本号由marjor-ver、minor-ver和build-number组成。
如响应示例中固件版本为1.5.79。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 285
{
  "result": 0,
  "file-path": "/data/new-firmware-6.mwf",
  "file-size": 14556364,
  "firmware-info": {
    "product-id": 770,
    "firmware-id": 0,
    "marjor-ver": 1,
    "minor-ver": 5,
    "build-number": 79,
    "build-time": "2021-05-13 05:27:59",
    "product-name": "Ultra Stream SDI",
    "firmware-name": "Standard",
    "hardware-ver": "A"
  }
}
```

add-firmware接口

将已上传的设备固件添加到有效固件列表中。

仅管理员有权限。

设备的固件通过uploadFirmware接口成功上传到云平台后，再执行本接口使固件生效，生效的固件才能被审核通过的同类型设备使用。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/firmware/add
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
file-path	上传到云平台的固件路径，接口uploadFirmware中返回的。

请求示例

```
POST /sc_cloud/firmware/add HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 40
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "file-path": "/data/new-firmware-6.mwf"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。1: 重复操作。其它: 参见API状态码。
method	方法名: add-firmware。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 36
{
  "method": "add-firmware",
  "result": 1
}
```

delete-firmware接口

删除云平台上的固件。支持批量删除。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/firmware/delete
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
ids	固件ID集合。通过接口 get-all-firmwares 获得。

请求示例

```
POST /sc_cloud/firmware/delete HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json; charset=utf-8
Content-Length: 12
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "ids": [24, 29]
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: add-firmware。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 36
{
  "method": "delete-firmware",
  "result": 0
}
```

get-users接口

获取云平台用户信息列表。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS GET
URL	ip[:port]/sc_cloud/scapi/get-users
数据格式	响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

请求示例

```
GET /sc_cloud/scapi/get-users HTTP/1.1
Accept: application/json, text/plain, */*
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
users	云平台用户信息列表。

users

参数	描述
id	用户ID。
name	用户名。Admin是 管理员 。
nick	昵称, 废弃。

管理员

- 用户名为Admin时, 表示是云平台管理员。
- 管理员是无法删除的。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 254
{
  "result": 0,
  "users": [
    {
      "id": 1,
      "name": "Admin",
      "nick": "Admin",
      "Passwd": "",
      "permit": -1
    },
    {
      "id": 2,
      "name": "Test",
      "nick": "",
      "Passwd": "",
      "permit": 1
    }
  ]
}
```


add-user接口

添加云平台用户。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/user/add
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
name	用户名。
passwd	密码, MD5加密。

请求示例

```
POST /sc_cloud/user/add HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 92
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "name": "guest",
  "passwd": "c4ca4238a0b923820dcc509a6f75849b"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
id	用户ID。
method	方法名: add-user。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 54
{
  "result": 0,
  "id": "5",
  "method": "add-user"
}
```

delete-user接口

删除云平台用户。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/user/delete
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	用户ID。

请求示例

```
POST /sc_cloud/user/delete HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json; charset=utf-8
Content-Length: 8
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYXPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 5
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: delete-user。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 35
{
  "method": "delete-user",
  "result": 0
}
```

change-passwd接口

云平台用户修改自己的登录密码，修改时必须输入原密码。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/user/change-passwd
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	用户ID。通过 用户登录 接口获得。
param	原密码，MD5加密。
passwd	新密码，MD5加密。

请求示例

```
POST /sc_cloud/user/change-passwd HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 95
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 1,
  "param": "e3afed0047b08059d0fada10f400c1e5",
  "passwd": "2e33a9b0b06aa0a01ede70995674ee23"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: change-passwd。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 37
{
  "method": "change-passwd",
  "result": 0
}
```

reset-passwd接口

重置云平台用户密码，无需输入原密码。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/user/reset-passwd
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	用户ID。通过 get-users 接口获得。
passwd	新密码，MD5加密。

请求示例

```
POST /sc_cloud/user/reset-passwd HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 52
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 2,
  "passwd": "c81e728d9d4c2f636f067f89cc14862c"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: reset-passwd。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 36
{
  "method": "reset-passwd",
  "result": 0
}
```


add-catalog接口

创建自定义的设备分组。

用户可以将设备[移动到](#)分组中，方便对设备进行分类管理。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/devcatalog/add-catalog
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	父分组ID。自定义的设备分组ID从1开始。
param	新分组名称。

请求示例

```
POST /sc_cloud/devcatalog/add-catalog HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json; charset=utf-8
Content-Length: 24
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 0,
  "param": "Group"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
id	新分组ID。
method	方法名: add-catalog。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 42
{
  "result": 0,
  "id": 7,
  "method": "add-catalog"
}
```

rename-catalog接口

修改自定义的设备分组名称。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/devcatalog/rename-catalog
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	需要重命名的设备分组ID。
param	分组的新名称。

请求示例

```
POST /sc_cloud/devcatalog/rename-catalog HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 28
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 7,
  "param": "Group_new"
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: rename-catalog。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 38
Connection: keep-alive
{
  "method": "rename-catalog",
  "result": 0
}
```

delete-catalog接口

删除自定义的设备分组。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/devcatalog/delete-catalog
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	需要删除的 设备分组ID 。

请求示例

```
POST /sc_cloud/devcatalog/delete-catalog HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 8
Connection: keep-alive
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 7,
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: delete-catalog。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 38
Connection: keep-alive
{
  "method": "delete-catalog",
  "result": 0
}
```

set-catalog接口

将指定设备移动到指定分组中。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/devcatalog/set-catalog
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	设备分组ID。通过 get-dev-catalogs 接口获得。 取值为0, 表示从自定义的设备分组移出。
params	批量移动到此分组的设备序列号集合。

请求示例

```
POST /sc_cloud/devcatalog/set-catalog HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json; charset=utf-8
Content-Length: 35
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 2,
  "params": [
    "A305200908004",
    "A304200731001"
  ]
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">0: 成功。其它: 参见API状态码。
method	方法名: set-catalog。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 35
{
  "method": "set-catalog",
  "result": 0
}
```


set-catalog-secu接口

自定义设备分组的安全管理设置。

仅管理员有权限。

接口原型

协议	HTTP/HTTPS POST
URL	ip[:port]/sc_cloud/devcatalog/set-catalog-secu
数据格式	请求消息: Content-Type: application/json 响应消息: Content-Type: application/json

请求Cookies

名称	描述
mwcloud-sid	用于标识当前登录的字符串
mwcloud-uid	当前登录用户名

Request Body

参数	描述
id	设备分组ID。
param-int2	常量值: 0。
param-int	分组是否所有用户可见。0: 否, 1: 是。
params	param-int取值为0时有效, 可见用户集合。

params

- param-int取值为1时, params取值为: [""]。
- param-int取值为0时, params为指定用户名的数组。

请求示例

```
POST /sc_cloud/devcatalog/set-catalog-secu HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json;charset=utf-8
Content-Length: 63
Cookie: mwcloud-sid=OCXWUMGEYPGIAWWOETYPNMVHDZIAZJP; mwcloud-uid=Admin;
{
  "id": 1,
  "param-int2": 0,
  "param-int": 0,
  // 本分组只有用户名为"Admin"和"Test"的用户可见
  "params": [
    "Admin",
    "Test"
  ]
}
```

响应参数

Response Header

参数	描述
Content-Type	取值为: 'application/json; charset=UTF-8'

Response Body

参数	描述
result	返回码。 <ul style="list-style-type: none">• 0: 成功。• 其它: 参见API状态码。
method	方法名: set-usercatalog。

响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 39
Connection: keep-alive
{
  "method": "set-usercatalog",
  "result": 0
}
```