

Z510 Media Agent API Spec

产品设计文档

Product Design Document (PDD)

Metis Station 教育市场家族系列

Revision 0.3.0

二月 2022

您不得将本文档与本文所述有关北京奕斯伟计算技术有限公司（以下简称“ESWIN”）产品的任何侵权或其他法律分析联系在一起使用或协助使用。您同意向 ESWIN 授予其后起草的任何专利权利要求的非排他性免版税许可，其中包括本文所披露的主题。

本文档不授予任何知识产权的许可（明示或暗示，禁止反言或其他方式）。

ESWIN 技术的功能和优势取决于系统配置，并且可能需要启用硬件，软件或服务才能激活。在 eswin.com 或从 OEM 或零售商处了解更多信息。

没有计算机系统可以绝对安全。ESWIN 对数据或系统的丢失或被盗或由此造成的任何损失不承担任何责任。

所描述的产品可能包含设计缺陷或错误，一经发现将收入勘误表，并因此可能导致产品与已出版的规格有所差异。可根据要求提供最新的勘误表。

ESWIN 不提供任何明示和暗示的保证，包括但不限于对适销性，针对特定目的的适用性和非侵权的暗示保证，以及由于性能，交易过程或交易使用而产生的任何保证。

ESWIN 技术的功能和优势取决于系统配置，并且可能需要启用硬件，软件或服务才能激活。从 eswin.com 或从 OEM 或零售商处了解更多信息。

此处提供的所有信息如有更改，恕不另行通知。请与您的 ESWIN 代表联系，以获取最新的 ESWIN 产品规格和路线图。

在下订单之前，请与您当地的 ESWIN 销售办事处或分销商联系，以获取最新规格。

ESWIN，ESWIN 徽标及 ESWIN 在中国和/或其他国家（地区）的商标。

*其他名称和品牌可能是其他所有者的财产。

版权所有©2021 ESWIN。版权所有。

Revision Number	Description	Author	Revision Date
0.1	•Initial Release.	卢睿	2021.11.19
0.2	•Add more interface to update video/audio task, set preview, set background, save snapshot, generate key frame and save media stream to file, etc.	卢睿	2021.12.15
0.3	•Switch MCS from hierarchical structure to flat structure. Generalize interface. Update parameter of some interface	卢睿	2022.02.22
	•		
	•		
	•		

目录

1 背景介绍.....	8
2 软件架构.....	8
3 Media Agent 接口	9
4 auth 接口	10
4.1 POST auth.....	10
4.2 POST auth/modify	11
5 mediadevice 接口	13
5.1 GET mediadevice/audio/in.....	13
5.2 GET mediadevice/audio/out.....	15
5.3 GET mediadevice/video/in	17
5.4 GET mediadevice/video/out.....	19
5.5 GET mediadevice/list.....	21
5.6 GET mediadevice/audio/in/capability	24
5.7 GET mediadevice/audio/out/capability	26
5.8 GET mediadevice/video/in/capability	28
5.9 GET mediadevice/video/out/capability	30
5.10 GET mediadevice/video/out/background.....	32
5.11 GET mediadevice/audio/in/configuration	34
5.12 GET mediadevice/audio/out/configuration.....	36
5.13 POST mediadevice/audio/in/configuration.....	37
5.14 POST mediadevice/audio/out/configuration.....	39
5.15 POST mediadevice/video/out/background.....	41
6 mediatask 接口	42
6.1 POST mediatask/create_update.....	43
6.1.1 AudioSpecs 说明.....	44
6.1.2 VideoSpecs 说明.....	45
6.1.3 CodecStreamSpecs 说明.....	45
6.1.4 AudioAiSpecs 说明.....	46
6.1.5 VideoAiSpecs 说明.....	46
6.1.6 VideoDirectorSpec 说明.....	47
6.1.7 AudioComposeSpecs 说明.....	48
6.1.8 VideoComposeSpecs 说明.....	48
6.1.9 VideoTransformSpecs 说明.....	49
6.1.10 PlaySpecs 说明.....	50
6.1.11 RenderSpecs 说明.....	50
6.1.12 AudioCodecSpecs 说明.....	51
6.1.13 VideoCodecSpecs 说明.....	52
6.1.14 RecordSpecs 说明.....	52
6.1.15 ServerSpecs 说明.....	53
6.1.16 MIOS.....	55
6.2 POST mediatask/remove	59
6.3 POST mediatask/suspend	60
6.4 POST mediatask/resume	62

6.5 POST mediatask/inspection.....	63
6.6 POST mediatask/keyframe.....	65
6.7 POST mediatask/snapshot.....	66
6.8 POST mediatask/merge.....	68
6.9 DELETE mediatask/{MCS_Name}.....	70
6.10 GET mediatask/tasks.....	71
6.11 GET mediatask/{MCS_Name}/{Spec_Name}.....	72

ESWIN ONLY FOR SUSON

表 1	Media Agent 接口请求公共结构	9
表 2	Media Agent 接口响应公共结构	9
表 3	返回 Code 列表及含义	10
表 4	POST auth 接口 Body 参数结构	10
表 5	POST auth 接口响应 Header 结构	11
表 6	POST auth/modify 接口 Body 参数结构	12
表 7	GET mediadevice/audio/in 接口响应 Body 结构	14
表 8	GET mediadevice/audio/out 接口响应 Body 结构	16
表 9	GET mediadevice/video/in 接口响应 Body 结构	18
表 10	GET mediadevice 接口响应 Body 结构	20
表 11	GET mediadevice/list 接口响应 Body 结构	22
表 12	GET mediadevice/audio/in/capability 接口响应 Body 结构	25
表 13	GET mediadevice/audio/out/capability 接口响应 Body 结构	27
表 14	GET mediadevice/video/in/capability 接口响应 Body 结构	29
表 15	GET mediadevice/video/out/capability 接口响应 Body 结构	31
表 16	GET mediadevice/video/out/background 接口响应 Body 结构	33
表 17	GET mediadevice/audio/in/configuration 接口响应 Body 结构	35
表 18	GET mediadevice/audio/configuration 接口响应 Body 结构	36
表 19	POST mediadevice/audio/in/configuration 接口 Body 参数结构	38
表 20	POST mediadevice/audio/out/configuration 接口 Body 参数结构	40
表 21	POST mediadevice/video/out/background 接口 Body 参数结构	41
表 22	MCS 一级结构	44
表 23	AudioSpecs 结构	44
表 24	VideoSpecs 结构	45
表 25	CodecStreamSpecs 结构	46
表 26	AudioAiSpecs 结构	46
表 27	VideoAiSpecs 结构	47
表 28	DirectorSpec 结构	48
表 29	AudioComposeSpecs 结构	48
表 30	VideoComposeSpecs 结构	49
表 31	VideoComposeSpecs 结构	50
表 32	PlaySpecs 结构	50
表 33	RenderSpecs 结构	51
表 34	AudioCodecSpecs 结构	52
表 35	VideoCodecSpecs 结构	52
表 36	RecordSpecs 结构	53
表 37	ServerSpecs 结构	53
表 38	POST mediatask/create 接口响应 Body 结构	54
表 39	ProcessSpecs 结构	59
表 40	POST mediatask/remove 接口 Body 参数结构	59
表 41	POST mediatask/suspend 接口 Body 参数结构	61
表 42	POST mediatask/resume 接口 Body 参数结构	62
表 43	POST mediatask/inspection 接口 Body 参数结构	64
表 44	POST mediatask/keyframe 接口 Body 参数结构	65

表 45 POST mediatask/snapshot 接口 Body 参数结构..... 67

表 46 POST mediatask/merge 接口 Body 参数结构..... 69

表 47 GET mediatask/tasks 接口响应 Body 结构..... 71

表 48 GET mediatask/tasks 接口响应 Body 结构..... 73

ESWIN ONLY FOR SUSON

1 背景介绍

为了与合作客户（以下简称“客户”）共同推动边缘计算的应用，ESWIN 为自研的 Metis Station 边缘计算站开发了具有普适性的多媒体处理和教育场景下的 AI 能力，并开放给客户。

ESWIN 定义了 Media Agent 来作为多媒体和 AI 算法等接口服务，为了降低客户的开发成本，接口设计方面具备以下特点：

- 使用 RESTful API 方式，无需客户程序编解或链接 ESWIN 提供的类库或运行环境；
- 命名简单，易于客户的工程师的理解和掌握；
- 参数均使用业界惯例的 JSON 数据结构；

Media Agent 中关于多媒体处理和 AI 服务的完整 JSON 数据结构全部在 Media Compose Spec（以下简称 MCS）。理解 MCS 的结构后，应用 MediaAgent 的能力将事半功倍。

本文的目的是描述 ESWIN 边缘设备所提供的多媒体业务相关的 RESTful API 接口规范，时序逻辑以及 MCS 等 JSON 结构，方便使用者设计自己的程序调用 RESTful API 接口完成其业务逻辑。下面将按这几方面分别展开描述。

2 软件架构

ESWIN Media Agent 的软件架构如下图所示：

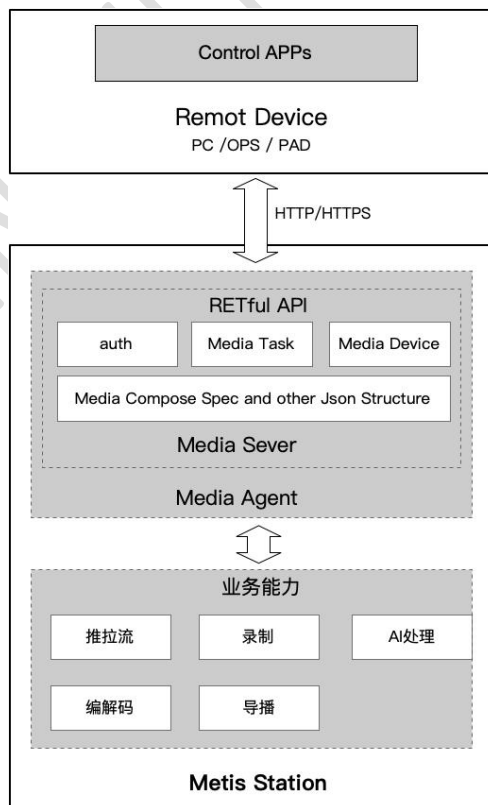


图 1 ESWIN Media Agent 软件架构示意图

整个架构由远程设备（如 OPS）上的一个 Control APP 和 ESWIN 边缘设备 Metis Station 上的 Media Agent 组成。Media Agent 以 RESTful API 的方式提供一系列访问接口，同时运行一个 HTTP Server，相应用户请求服务。Control APP 通过 HTTP 连接到 Media Agent 的 HTTP Server，访问 Media Agent 提供的 RESTful API 请求相应的多媒体服务。为了确保私密，Media Agent 提供更安全的 HTTP 连接，即 HTTPS 连接。

Media Agent 的接口分为 auth，mediadevice 和 mediatask 三类，auth 类接口用于向 Media Agent 认证，取得进一步操作的令牌；mediadevice 类接口用于获取 ESWIN 边缘设备上多媒体子系统的相关信息及设置；mediatask 类的接口则用于完成启动，结束和管理多媒体及 AI 处理的任务。三类接口包含有 GET 和 POST 两种请求，其中 GET 请求用于查询，POST 请求用于创建，修改，删除一项任务或修改一项配置。

3 Media Agent 接口

接口请求公共部分

Media Agent 的接口请求和响应包含一些公共部分。对于接口请求而言，除了 POST system/auth 接口请求外，所有的接口请求应当在其请求的 Header 段中包含一个 Token 键值，以标识用户取得的令牌信息。

键名	段名	说明
Token	Header	必须，字符串，用户取得的令牌信息。

表 1 Media Agent 接口请求公共结构

接口响应公共部分

Media Agent 的接口调用均会通过 Body 段返回一个 JSON 结构的响应。该 JSON 结构有一部分为所有响应均会包含的公共部分。此外，响应的 Header 段中还可能包含一个 Token 键值，用以通知用户后续操作需要使用的令牌信息。接口响应的公共部分描述如下：

字段/键名	段名	说明
Token	Header	可选，字符串，令牌信息。
Code	Body	必须，HTTP 返回代码。 • 2xx 表示成功 • 4xx 和 5xx 均表示失败（注，4xx 返回码与 Http 常见返回码不同）
Message	Body	必须，对应于返回代码可读的文字描述。

表 2 Media Agent 接口响应公共结构

示例

```
{
  "Code": 200,
  "Message": "No error found!"
}
{
  "Code": 400,
```

```
"Message": "Bad Parameter!"
}
```

接口响应 Code 列表

下表给出了所有接口响应可能返回的 Code 列表及其含义。

Code	含义
200	接口调用正确，没有错误。
201	Media task 创建成功。
400	接口调用时的参数错误。
403	接口调用时的 token 无效。
404	接口调用时的 TaskId 错误。
409	资源冲突导致任务创建失败。
500	Media Agent 内部错误，无法提供服务。

表 3 返回 Code 列表及含义

4 auth 接口

auth 类接口包括以下几个接口：

- POST auth，用于向 Media Agent 认证，获得令牌信息。
- POST auth/modify，用来更新用户名及口令信息。

4.1 POST auth

功能说明

用于向 Media Agent 认证，以获得后续操作的令牌。

参数说明

路径

无

Header

无

Body

JSON 结构，描述如下：

字段名	说明
UserName	必须，字符串，用户名。初始用户为“admin”。
Password	必须，字符串，用户密码。初始密码为“admin”。

表 4 POST auth 接口 Body 参数结构

响应说明

Header

键值	说明
Token	可选，字符串，只有当 Body 中的 Code 为 200 时包含，令牌信息，后续接口调用需要使用该令牌。

表 5 POST auth 接口响应 Header 结构

Body

除公共字段外无其他字段。

示例

请求

POST auth

Header

无

Body

```
{
  "UserName": "admin",
  "Password": "admin"
}
```

响应

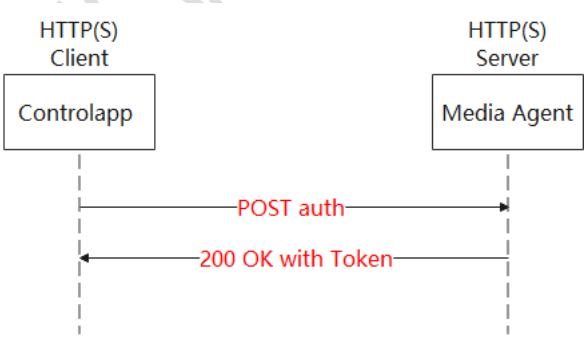
Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "Code": 200,
  "Message": "No error found!",
}
```

时序



4.2 POST auth/modify

功能说明

用于向 Media Agent 更新认证用户名及密码。

参数说明

路径

无

Header

Token:{token}

Body

JSON 结构，除公共部分外的结构如下：

字段名	说明
NewName	必须，字符串，新用户名。
NewPassword	必须，字符串，新用户密码。

表 6 POST auth/modify 接口 Body 参数结构

响应说明

Header

可能包含更新的 Token 键值。

Body

除公共字段外无其他字段。

示例

请求

POST auth/modify

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "NewName" : "user",
  "NewPassword" : "a123456."
}
```

响应

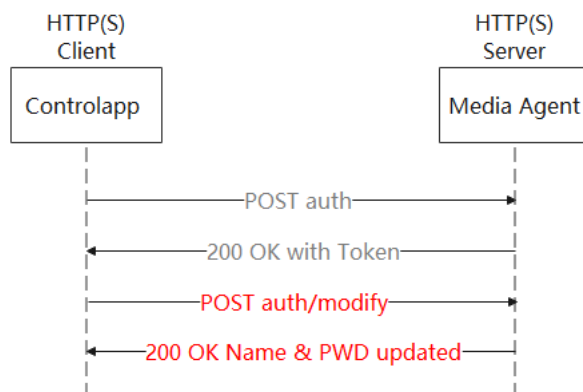
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
}
```

时序



5 mediadevice 接口

mediadevice 类接口用于查询边缘设备上的多媒体子系统信息并进行必要的控制。该类接口包括：

- GET mediadevice/audio/in, 获取音频输入子系统信息。
- GET mediadevice/audio/out, 获取音频输出子系统信息。
- GET mediadevice/video/in, 获取视频输入子系统信息。
- GET mediadevice/video/out, 获取视频输出子系统信息。
- GET mediadevice/audio/in/capability, 获取音频输入子系统能力。
- GET mediadevice/audio/out/capability, 获取音频输出子系统能力。
- GET mediadevice/video/in/capability, 获取视频输入子系统能力。
- GET mediadevice/video/out/capability, 获取视频输出子系统能力。
- GET mediadevice/video/out/background, 获取视频输出设备的背景设置。
- GET mediadevice/audio/in/volume, 获取音频输入子系统音量。
- GET mediadevice/audio/out/volume, 获取音频输出子系统音量。
- POST mediadevice/audio/in/volume, 设置音频输入子系统音量。
- POST mediadevice/audio/out/volume, 设置音频输出子系统音量。
- POST mediadevice/video/out/background, 设置视频输出设备的背景。

5.1 GET mediadevice/audio/in

功能说明

用于获取边缘设备上音频输入子系统信息。

参数说明

路径

无

Header

Token:{token}

Body
无

响应说明

Header
可能包含更新的 Token 键值。

Body
JSON 结构除公共字段外的结构如下：

字段名	说明
AudioIns	必须，结构数组，音频输入设备列表。
Deviceld	必须，字符串，音频输入设备系统 ID。
DeviceName	必须，字符串，音频输入设备系统名。
AliasName	可选，字符串，音频输入设备的别名。
CanLocalAmplification	可选，布尔值，音频输入设备是否支持本地扩声。
Default	必须，布尔值，是否是默认音频输入设备。仅有一个设备该字段设定为“true”。
SubDevices	可选，结构数组，为该音频输入设备下的子设备列表。
Id	必须，字符串，子设备 ID。
Name	必须，字符串，子设备名。
Alias	可选，字符串，子设备别名。
Algorithms	可选，结构数组，为该音频输入设备配置的 AI 算法参数。
Name	必须，字符串，AI 算法名。
Default	可选，布尔值，是否是默认 AI 算法。仅有一个算法该字段设定为“true”。
Format	可选，字符串，该 AI 算法支持的输出数据流格式。如“Audio”，“Data”，“Audio+Data”。

表 7 GET mediadevice/audio/in 接口响应 Body 结构

示例
请求

GET mediadevice/audio/in

参数

Header
Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzO
TA0NJyYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body
无

响应

Header

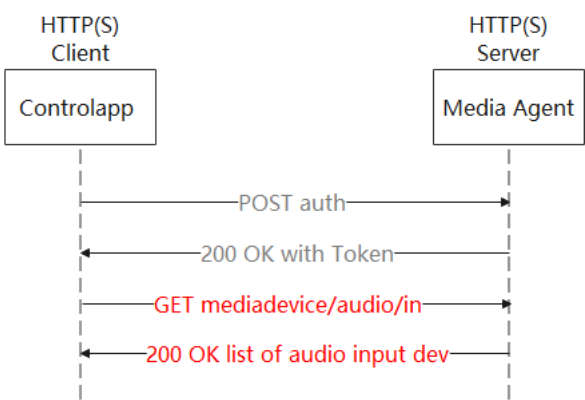
无

Body

```
{
  "Code": 200,
  "Message": "No error found!",

  "Audiolns": [
    {
      "DeviceId": "0102-0200",
      "DeviceName": "Built-in Audio Capture Stereo",
      "AliasName": "Default Microphone",
      "CanLocalAmplification": true,
      "Default": true,
      "SubDevices": [
        {
          "Id": "0000",
          "Name": "Main Micphone",
          "Alias": "Fixed Mic"
        },
        {
          "Id": "0001",
          "Name": "Aux Micphone",
          "Alias": "Wireless Mic"
        }
      ]
    }
  ]
}
```

时序



5.2 GET mediadevice/audio/out

功能说明

用于获取边缘设备上音频输出子系统信息。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构除公共字段外的结构如下：

字段名	说明
AudioOuts	必须，结构数组，音频输出设备列表。
DeviceId	必须，字符串，音频输出设备系统 ID。
DeviceName	必须，字符串，音频输出设备系统名。
AliasName	可选，字符串，视频输出设备的别名。
Default	必须，布尔值，是否是默认音频输出设备。仅有一个设备该字段设定为“true”。
SubDevices	可选，结构数组，为该音频输出设备下的子设备列表。
Id	必须，字符串，子设备 ID。
Name	必须，字符串，子设备名。
Alias	可选，字符串，子设备别名。

表 8 GET mediadevice/audio/out 接口响应 Body 结构

示例

请求

GET mediadevice/audio/out

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWVRhLTQ4ZjltOGZhYi1jZWYzO
TA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

Header

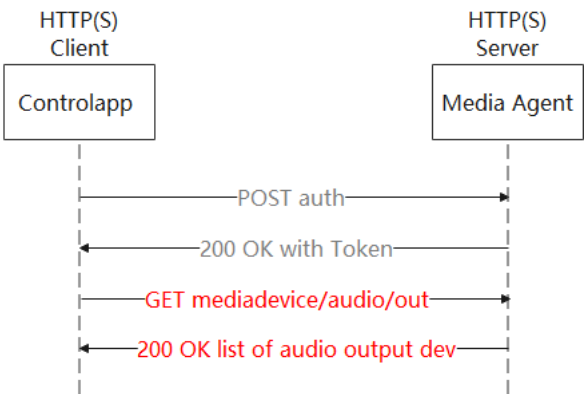
无

Body

{
 "Code": 200,


```
"Message": "No error found!",
"AudioOuts": [
  {
    "DeviceName": "Built-in Audio Playback Stereo",
    "DeviceId": "0202-0200",
    "AliasName": "Default Speaker",
    "Default": true,
    "SubDevices": [
      {
        "Id": "0000",
        "Name": "Main Speaker",
        "Alias": "Fixed Speaker"
      },
      {
        "Id": "0001",
        "Name": "Aux Speaker",
        "Alias": "Wireless Speaker"
      }
    ]
  }
]
```

时序



5.3 GET mediadevice/video/in

功能说明

用于获取边缘设备上视频输入子系统信息。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明**Header**

可能包含更新的 Token 键值。

Body

JSON 结构除公共字段外的结构如下：

字段名	说明
Videos	必须，结构数组，视频输入设备列表。
DeviceName	必须，字符串，视频输入设备系统名。
DeviceId	必须，字符串，视频输入设备系统 ID。
AliasName	可选，字符串，视频输入设备的别名。
Default	可选，布尔值，是否是默认视频输入设备。仅有一个设备该字段设定为“true”。
Algorithms	可选，字符串列表，视频输入设备配置的 AI 算法名。
Name	必须，字符串，AI 算法名。
Default	可选，布尔值，是否是默认 AI 算法。仅有一个算法该字段设定为“true”。
Format	可选，字符串，该 AI 算法支持的输出数据流格式。如“Video”，“Data”，“Video+Data”。

表 9 GET mediadevice/video/in 接口响应 Body 结构

示例**请求**

GET mediadevice/video/in

参数**Header**

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTAwNjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应**Header**

无

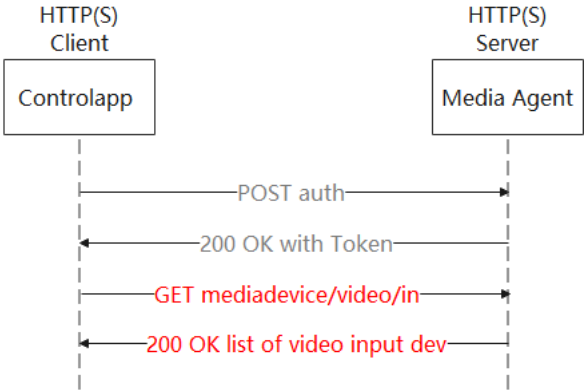
Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
  "Videos" : [
    {
      "DeviceName" : "HDMI_CAM_0",
      "DeviceId" : "0400-0000",
      "AliasName" : "Teacher",
      "Default" : true,
```

```

    "Algorithms" : [
      {
        "Name" : "ALGORITHM_TEACHER_TRACK",
        "Default" : true
      }
    ]
  }
  {
    "DeviceName" : "HDMI_CAM_1",
    "DeviceId" : "0400-0001",
    "AliasName" : "Student",
    "Default" : false,
    "Algorithms" : [
      {
        "Name" : "ALGORITHM_STUDENT_TRACK",
        "Default" : true
      }
    ]
  }
]
}
```

时序



5.4 GET mediadevice/video/out

功能说明

用于获取边缘设备上视频输出子系统信息。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构除公共字段外的结构如下：

字段名	说明
VideoOuts	必须，结构数组，视频输出设备列表。
DeviceId	必须，字符串，视频输出设备系统 ID。
DeviceName	必须，字符串，视频输出设备系统名。
AliasName	可选，字符串，视频输出设备的别名。
Default	必须，布尔值，是否是默认视频输出设备。仅有一个设备该字段设定为“true”。
Width	必须，整数，视频输出设备分辨率宽度。
Height	必须，整数，视频输出设备分辨率高度。

表 10 GET mediadevice 接口响应 Body 结构

示例

请求

GET mediadevice/video/out

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

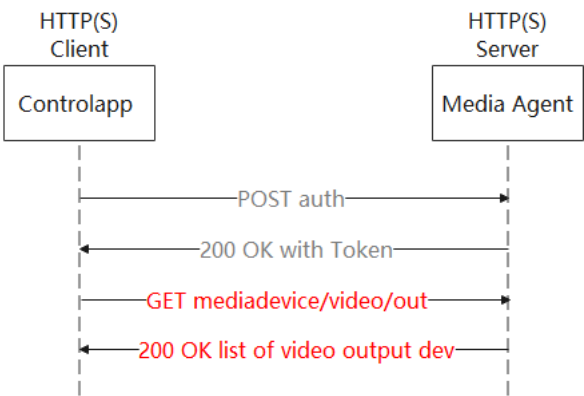
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
  "VideoOuts" : [
    {
      "DeviceName" : "HDMI-0",
      "DeviceId" : "0800-0002",
      "AliasName" : "Preview_Screen",
      "Default" : true,
      "Width" : 1920,
      "Height" : 1080
    }
  ]
}
```

时序



5.5 GET mediadevice/list

功能说明

用于获取边缘设备上所有音，视频输入子系统信息。该接口调用相当于一次调用 GET mediadevice/audio/in， GET mediadevice/audio/out， GET mediadevice/video/in， GET mediadevice/video/out 四个接口。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构除公共字段外的结构如下：

字段名	说明
Audiolns	必须， 结构数组， 音频输入设备列表。结构详情见 接口 GET mediadevice/audio/in 返回结构。
AudioOuts	必须， 结构数组， 音频输出设备列表。结构详情见 接口 GET mediadevice/audio/out 返回结构。
Videolns	必须， 结构数组， 视频输入设备列表。结构详情见 接口 GET mediadevice/video/in 返回结构。

VideoOuts	必须，结构数组，视频输出设备列表。结构详情见 接口 GET mediadevice/video/out 返回结构。
-----------	---------------------------------------------------------------------------

表 11 GET mediadevice/list 接口响应 Body 结构

示例
请求

GET mediadevice/list

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

Header

无

Body

```
{
  "Code": 200,
  "Message": "No error found!",

  "AudioIns": [
    {
      "DeviceId": "0102-0200",
      "DeviceName": "Built-in Audio Capture Stereo",
      "AliasName": "Default Microphone",
      "CanLocalAmplification": true,
      "Default": true,
      "SubDevices": [
        {
          "Id": "0000",
          "Name": "Main Micphone",
          "Alias": "Fixed Mic"
        },
        {
          "Id": "0001",
          "Name": "Aux Micphone",
          "Alias": "Wireless Mic"
        }
      ]
    }
  ],

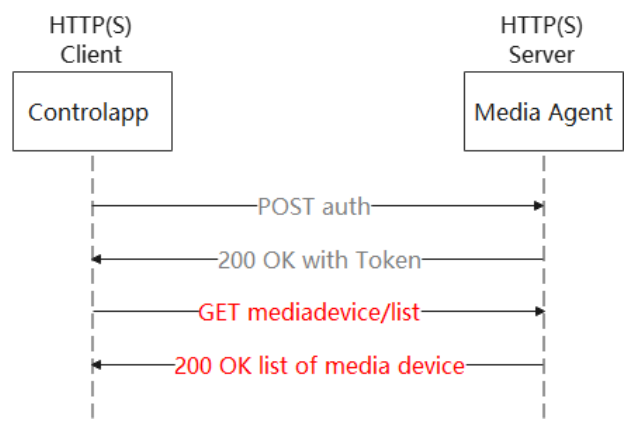
  "AudioOuts": [
    {
      "DeviceName": "Built-in Audio Playback Stereo",
      "DeviceId": "0202-0200",
      "AliasName": "Default Speaker",
      "Default": true,
      "SubDevices": [
        {
```

```

        "Id": "0000",
        "Name": "Main Speaker",
        "Alias": "Fixed Speaker"
    },
    {
        "Id": "0001",
        "Name": "Aux Speaker",
        "Alias": "Wireless Speaker"
    }
]
},
],
"VideoIns": [
    {
        "DeviceName": "HDMI_CAM_0",
        "DeviceId": "0400-0000",
        "AliasName": "Teacher",
        "Default": true,
        "Algorithms": [
            {
                "Name": "ALGORITHM_TEACHER_TRACK",
                "Default": true
            }
        ]
    }
    {
        "DeviceName": "HDMI_CAM_1",
        "DeviceId": "0400-0001",
        "AliasName": "Student",
        "Default": false,
        "Algorithms": [
            {
                "Name": "ALGORITHM_STUDENT_TRACK",
                "Default": true
            }
        ]
    }
],
"VideoOuts": [
    {
        "DeviceName": "HDMI-0",
        "DeviceId": "0800-0002",
        "AliasName": "Preview_Screen",
        "Default": true,
        "Width": 1920,
        "Height": 1080
    }
]
}

```

时序



5.6 GET mediadevice/audio/in/capability

功能说明

获取边缘设备音频采集的能力列表。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构，除公共字段外的结构如下：

字段名	说明
Audiolns	必须，结构数组，音频采集能力列表。每个元素均为一个音频输入设备的相关能力描述。
DeviceId	必须，字符串，音频输入设备 ID。
ChannelsRange	必须，两个元素的整数数组，音频输入设备支持的可能通道数，如 [1, 2]。
SampleRateRange	必须，两个元素的整数数组，音频输入设备支持的可能采样频率，如[8000, 16000, 32000, 44100, 48000]，单位是 hz。
Formats	必须，字符串数组，音频输入设备支持的采样格式列表，如“S16_LE”
Layouts	必须，字符串数组，音频采样帧不同 channel 的数据样本的组合方

	式列表，如“Interleaved”。
CurrentConfig	可选，结构，该音频输入设备当前配置的采集参数。
Channels	必须，整数，音频输入设备配置的采集通道数。
SampleRate	必须，整数，音频输入设备配置的采样率。
Format	必须，字符串，音频输入设备配置的采样格式。
Layout	必须，字符串，音频输入设备配置的数据样本的组合方式。
Codecs	必须，结构数组，每个元素为边缘设备支持的一种音频编码标准。
Standard	必须，字符串，音频编码标准名称，当前支持“opus”，“aac”。
Default	必须，布尔值，是否是默认音频编码标准。仅有一个标准该字段设定为“true”。

表 12 GET mediadevice/audio/in/capability 接口响应 Body 结构

示例
请求

GET mediadevice/audio/in/capability

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

Header

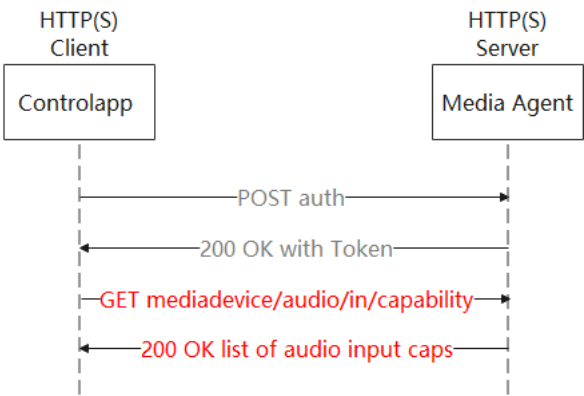
无

Body

```
{
  "Code": 200,
  "Message": "No error found!",
  "AudioIns": [
    {
      "DeviceId": "0102-0200",
      "ChannelsRange": [1, 2],
      "SampleRateRange": [8000, 16000, 32000, 44100, 48000],
      "Formats": ["S16_LE"],
      "Layouts": ["Interleaved"],
      "CurrentConfig": {
        "Channels": 2,
        "SampleRate": 48000,
        "Format": "S16_LE",
        "Layout": "Interleaved"
      }
    }
  ],
  "Codecs": [
```

```
{
  {
    "Standard" : "opus",
    "Default" : true,
  },
  {
    "Standard" : "aac",
    "Default" : false,
  }
}
```

时序



5.7 GET mediadevice/audio/out/capability

功能说明

获取边缘设备音频播放的能力列表。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构，除公共字段外的结构如下：

字段名	说明
AudioOuts	必须，结构数组，音频播放能力列表。每个元素均为一个音频输出

	设备的相关能力描述。
DeviceId	必须，字符串，音频输出设备 ID。
Codecs	必须，结构数组，每个元素为边缘设备支持的一种音频解码标准。
Standard	必须，字符串，音频解码标准名称，当前支持“opus”，“aac”。

表 13 GET mediadevice/audio/out/capability 接口响应 Body 结构

示例

请求

GET mediadevice/audio/out/capability

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

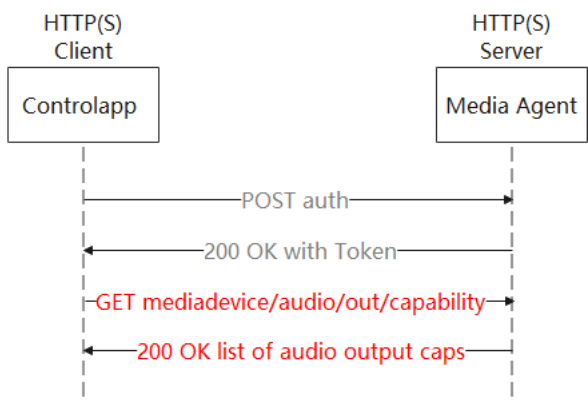
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!",
  "AudioOuts" : [
    {
      "DeviceId" : "0202-0200"
    }
  ],
  "Codecs" : [
    {
      "Standard" : "opus",
      "Default" : true,
    },
    {
      "Standard" : "aac",
      "Default" : false,
    }
  ]
}
```

时序



5.8 GET mediadevice/video/in/capability

功能说明

获取边缘设备视频采集的能力列表。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构，除公共字段外的结构如下：

字段名	说明
VideosIns	必须，结构数组，视频采集能力列表。每个元素均为一个视频输入设备的相关能力描述。
DeviceId	必须，字符串，视频输入设备 ID。
FormatSet	必须，结构数组，视频输入设备支持格式集合。
Format	必须，字符串，视频输入设备支持的采样格式，如“UYVY”，“MJPEG”，“h.264”。
ResolutionSet	必须，数组，视频输入设备对应当前采样格式支持的分辨率和对应的帧率列表。
Width	必须，整数，视频输入设备支持的分辨率宽度，如 1920。
Height	必须，整数，视频输入设备支持的分辨率高度，如 1080。

FrameRates	必须，整数数组，视频输入设备在该分辨率下支持的帧率列表，如 [60]。
CurrentConfig	可选，结构，该视频输入设备当前配置的采集参数。
Format	必须，整数，视频输入设备配置的采集格式。
Width	必须，整数，视频输入设备配置的采集分辨率宽度。
Height	必须，整数，视频输入设备配置的采集分辨率高度。
FrameRate	必须，整数，视频输入设备配置的采集帧率。
Codecs	必须，结构数组，每个元素为边缘设备支持的一种视频编码标准。
Standard	必须，字符串，视频编码标准名称，当前支持“h.264”，“h.265”，“vp9”。
Default	必须，布尔值，是否是默认视频编码标准。仅有一个标准该字段设定为“true”。

表 14 GET mediadevice/video/in/capability 接口响应 Body 结构

示例

请求

GET mediadevice/video/in/capability

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWVRhLTQ4ZjltOGZhYi1jZWYzOTAwNjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

Header

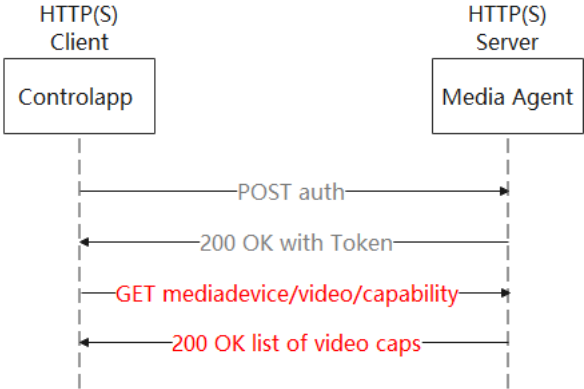
无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!",
  "VideoIns" : [
    {
      "DeviceId" : "0400-0000",
      "FormatSet" : [
        {
          "Format" : "UYVY",
          "ResolutionSet" : [
            {
              "Width" : 1920,
              "Height" : 1080,
              "FrameRates" : [30, 60]
            },
            {
              "Width" : 1280,
              "Height" : 720,
              "FrameRates" : [30, 60]
            }
          ]
        }
      ]
    }
  ]
}
```

```
    ]
  }
]
"CurrentConfig" :
{
  "Format" : "UYVY",
  "Width" : 1920,
  "Height" : 1080,
  "FrameRate" : 30
}
}
]
"Codecs" : [
{
  "Standard" : "h.264",
  "Default" : true,
},
{
  "Standard" : "h.265",
  "Default" : false,
},
{
  "Standard" : "vp9",
  "Default" : false,
}
]
]
```

时序



5.9 GET mediadevice/video/out/capability

功能说明

获取边缘设备视频输出的能力列表。

参数说明

路径

无

Header

Token:{token}

Body
无

响应说明

Header
可能包含更新的 Token 键值。

Body
JSON 结构，除公共字段外的结构如下：

字段名	说明
VideoOuts	必须，结构数组，视频播放能力列表。每个元素均为一个视频输出设备的相关能力描述。
DeviceId	必须，字符串，视频输出设备 ID。
Width	必须，整数，视频输出设备支持的分辨率宽度，如 1920。
Height	必须，整数，视频输出设备支持的分辨率高度，如 1080。
Codecs	必须，结构数组，每个元素为边缘设备支持的一种视频解码标准。
Standard	必须，字符串数组，边缘设备支持的视频解码格式。当前支持“h.264”，“h.265”，“vp9”。

表 15 GET mediadevice/video/out/capability 接口响应 Body 结构

示例
请求

GET mediadevice/video/out/capability

参数

Header
Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body
无

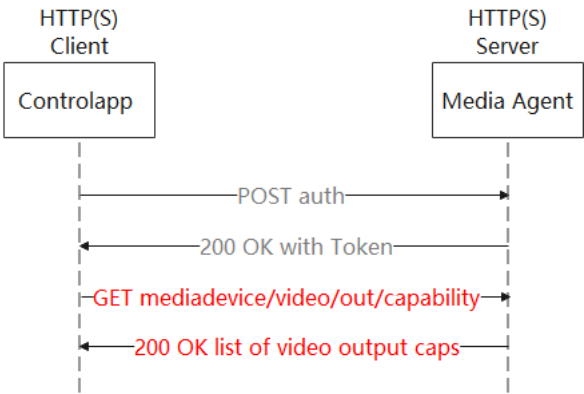
响应

Header
无

Body
{
 "Code" : 200,
 "Message" : "No error found!",
 "VideoOuts" : [
 {
 "DeviceId" : "0800-0002",
 "Width" : 1920,
 "Height" : 1080

```
    }
  ]
  "Codecs" : [
    {
      "Standard" : "h.264",
      "Default" : true,
    },
    {
      "Standard" : "h.265",
      "Default" : false,
    },
    {
      "Standard" : "vp8",
      "Default" : false,
    },
    {
      "Standard" : "vp9",
      "Default" : false,
    }
  ]
}
```

时序



5.10 GET mediadevice/video/out/background

功能说明

获取边缘设备视频输出子系统的永久背景设置。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构，除公共字段外的结构如下：

字段名	说明
VideoOuts	必须，结构数组，视频输出设备背景信息列表。
DeviceId	必须，字符串，视频输出设备 ID。
Background	必须，字符串，该视频输出设备的背景图片文件名（含路径）。“null”表示无背景。

表 16 GET mediadevice/video/out/background 接口响应 Body 结构

示例

请求

GET mediadevice/video/out/background

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzO
TA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

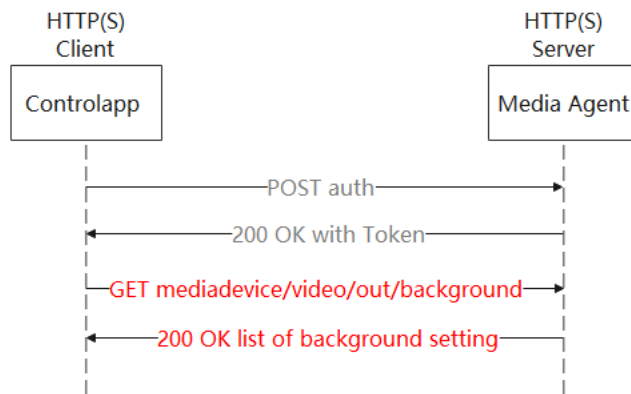
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!",
  "VideoOuts" : [
    {
      "DeviceId" : "0800-0002",
      "Background" : "/usr/share/backgrounds/eswin_wallpaper_1080.png",
    }
  ]
}
```

时序



5.11 GET mediadevice/audio/in/configuration

功能说明

获取边缘设备音频输入子系统的永久配置信息。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构，除公共字段外的结构如下：

字段名	说明
Audiolns	必须，结构数组，音频输入设备音量列表。
DeviceId	必须，字符串，音频输入设备 ID。
Mute	必须，布尔值，当前是否静音。即操作系统中显示的该设备静音设置。
Volume	必须，整数，范围[0-100]，音频输入设备的当前音量设置。即操作系统中显示的该设备音量。
LocalAmplification	必须，布尔值，音频输入设备是否需要本地扩声。注意：若该设备不支持本地扩声，则该字段值也为“false”。
SubDevices	可选，结构数组，为该音频输入设备下的子设备音量列表。

Id	必须，字符串，子设备 ID。
Mute	必须，布尔值，子设备是否静音。
Volume	必须，整数，范围[0-100]，子设备的当前音量设置。

表 17 GET mediadevice/audio/in/configuration 接口响应 Body 结构

示例
请求

GET mediadevice/audio/in/configuration

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

Header

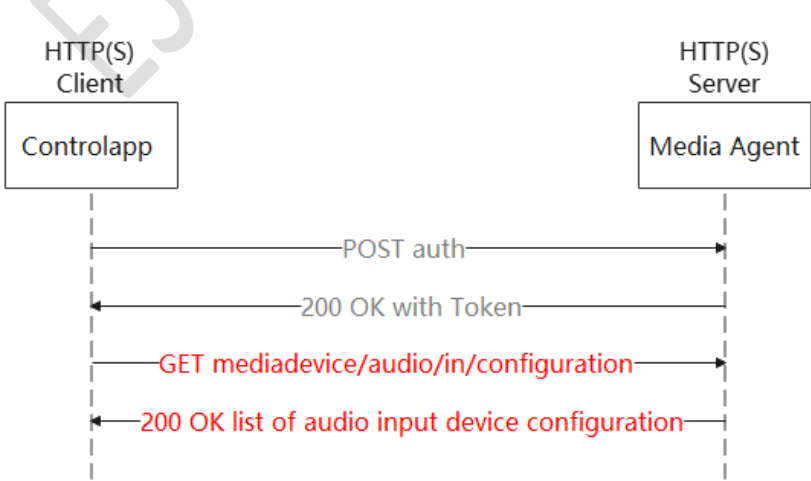
无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!",

  "AudioIns" : [
    {
      "DeviceId" : "0102-0200",
      "Mute" : false,
      "Value" : 50
    }
  ]
}
```

时序



5.12 GET mediadevice/audio/out/configuration

功能说明

获取边缘设备音频输出子系统的永久配置信息。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构，除公共字段外的结构如下：

字段名	说明
AudioOuts	必须，结构数组，音频输出设备音量列表。
DeviceId	必须，字符串，音频输出设备 ID。
Mute	必须，布尔值，当前是否静音。即操作系统中显示的该设备静音设置。
Volume	必须，整数，范围[0-100]，音频输出设备的当前音量设置。即操作系统中显示的该设备音量。
SubDevices	可选，结构数组，为该音频输出设备下的子设备音量列表。
Id	必须，字符串，子设备 ID。
Mute	必须，布尔值，子设备是否静音。
Volume	必须，整数，范围[0-100]，子设备的当前音量设置。

表 18 GET mediadevice/audio/configuration 接口响应 Body 结构

示例

请求

GET mediadevice/audio/out/configuration

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTAwNjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

Header

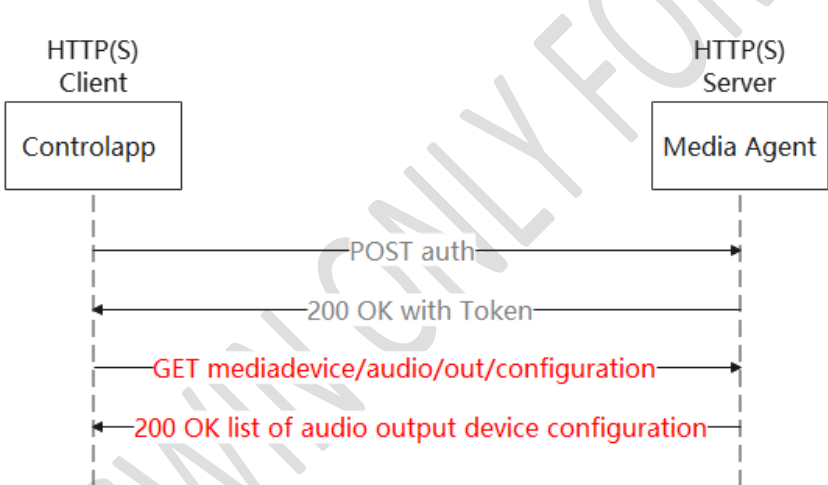
无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!",

  "AudioOuts" : [
    {
      "DeviceId" : "0202-0200",
      "Mute" : false,
      "Value" : 50
    }
  ]
}
```

时序



5.13 POST mediadevice/audio/in/configuration

功能说明

用于更新边缘设备音频输入子系统的永久设置。

参数说明

路径

无

Header

Token:{token}

Body

JSON 结构，详细描述如下：

字段名	说明
DeviceId	必须，字符串，音频输入设备 ID。
Mute	可选，布尔值，开关静音。修改后操作系统中显示的该设备静音设置也将被更新。
Volume	可选，整数，仅当“Mute”不指定或为 false 时使用，为该设备的设置的最新音量。范围[0-100]。修改后操作系统中显示的该设备音量也将被更新。
LocalAmplification	可选，布尔值，音频输入设备是否需要本地扩声。注意：若该设备不支持本地扩声，则该字段设置不生效。
SubDevices	可选，结构数组，为该音频输入设备下的子设备音量设置列表。
Id	必须，字符串，子设备 ID。
Mute	可选，布尔值，子设备是否需要静音。
Value	可选，整数，范围[0-100]，子设备的目标音量设置。仅当“Mute”不指定或为 false 时使用。

表 19 POST mediadevice/audio/in/configuration 接口 Body 参数结构

响应说明

Header

可能包含更新的 Token 键值。

Body

除公共字段外无其他字段。

示例

请求

POST mediadevice/audio/in/configuration

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzO
TA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "DeviceId" : "0102-0200",
  "Value" : 60
}
```

响应

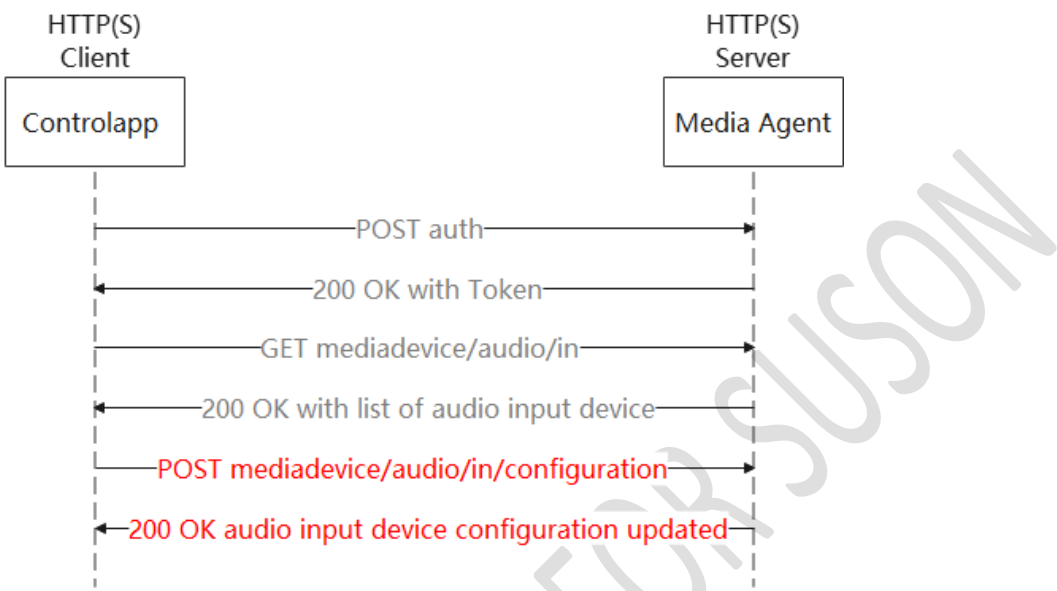
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
}
```

时序



5.14 POST mediadevice/audio/out/configuration

功能说明

用于更新边缘设备音频输出子系统的永久设置。

参数说明

路径

无

Header

Token:{token}

Body

JSON 结构，详细描述如下：

字段名	说明
DeviceId	必须，字符串，音频输出设备 ID。
Mute	可选，布尔值，是否静音。修改后操作系统中显示的该设备静音设置也将被更新。
Value	可选，整数，仅当“Mute”不指定或为 false 时使用，为该设备的设置的最新音量，范围[0-100]。修改后操作系统中显示的该设备音量也将被更新。

SubDevices	可选，结构数组，为该音频输出设备下的子设备音量设置列表。
Id	必须，字符串，子设备 ID。
Mute	可选，布尔值，子设备是否需要静音。
Value	可选，整数，范围[0-100]，子设备的目标音量设置。仅当“Mute”不指定或为 false 时使用，范围[0-100]。

表 20 POST mediadevice/audio/out/configuration 接口 Body 参数结构

响应说明

Header

可能包含更新的 Token 键值。

Body

除公共字段外无其他字段。

示例

请求

POST mediadevice/audio/out/configuration

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWVRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "DeviceId" : "0202-0200",
  "Value" : 30
}
```

响应

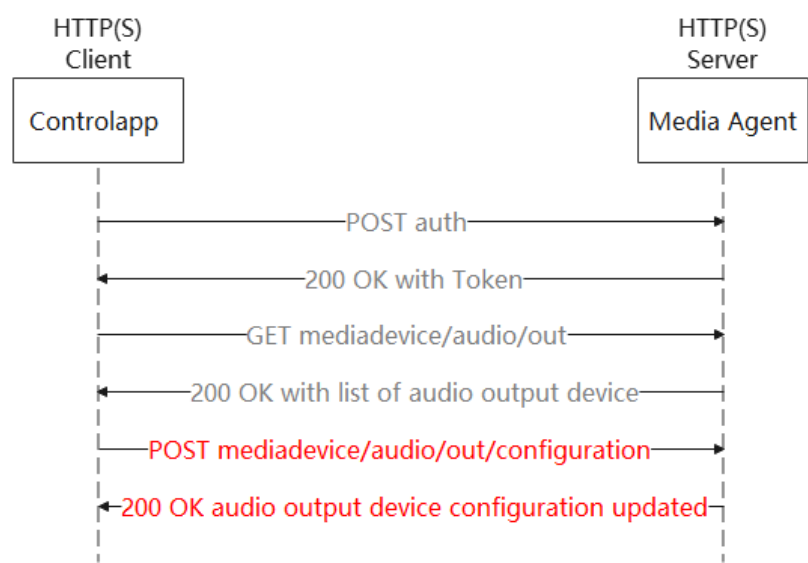
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
}
```

时序



5.15 POST mediadevice/video/out/background

功能说明
用于更新边缘设备视频输出子系统的永久背景设置。

参数说明
路径
无

Header
Token:{token}

Body
JSON 结构，详细描述如下：

字段名	说明
DeviceId	必须，字符串，视频输出设备 ID。
Background	必须，字符串，该回显位置背景图片文件名。当文件名中不包含路径时，使用系统路径“/usr/share/backgrounds”。使用“null”表示无背景，支持 jpeg, png 格式的文件。

表 21 POST mediadevice/video/out/background 接口 Body 参数结构

响应说明
Header
可能包含更新的 Token 键值。

Body
除公共字段外无其他字段。

示例

请求

```
POST mediadevice/video/out/background
```

参数

Header

```
Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM
```

Body

```
{
  "DeviceId": "0800-0002",
  "Background": "black.png"
}
```

响应

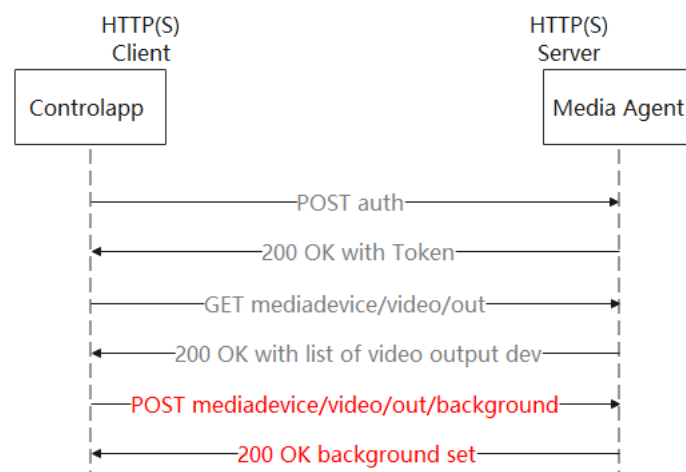
Header

无

Body

```
{
  "Code": 200,
  "Message": "No error found!"
}
```

时序



6 mediatask 接口

mediatask 类接口用于创建，删除多媒体及 AI 任务并进行必要的运行时控制。该类接口包括：

- POST mediatask/create_update，创建或更新一系列多媒体任务。
- POST mediatask/remove，销毁某个多媒体任务。

- POST mediatask/suspend, 暂停某个多媒体任务的执行。
- POST mediatask/resume, 恢复某个多媒体任务的执行。
- POST mediatask/inspection, 获取某个任务系列的完整 MCS 描述。
- POST mediatask/keyframe, 触发指定视频编码任务立即生成一个 IDR 帧。
- POST mediatask/snapshot, 从指定的视频流中捕获一个快照。
- DELETE mediatask/{MCS_Name}, 销毁一个之前用 MCS 创建的多媒体任务系列。
- GET mediatask/tasks, 获得用户当前运行的多媒体任务系列列表。
- GET mediatask/{MCS_Name}/{Spec_Name}, 获得指定多媒体任务的详细状态。

6.1 POST mediatask/create_update

功能说明

该接口可以用于创建或更新一系列多媒体任务, 包括采集, 进行 AI 处理, 本地录制, 导播, 推拉流, 播放和回显等。接口参数 JSON 结构中的根“Name”字段标识这一多媒体任务系列。每一个子任务由对应 Spec 中的“Name”字段标识。相同的根“Name”标识相同的任务系列, 其各个 Spec 之间可以相互引用。位于同一任务系列中的同名 Spec 标识相同的多媒体任务, 当某个 Spec “Name”第一次出现时, MA 将创建对应的多媒体任务。后续出现的同一系列中的同名任务将被视为对于原有任务的更新。

需要注意的是只有部分的 Spec 字段可以被更新, 支持动态更新的字段已经在各 Spec 的表格中给出, 请参阅各 Spec 表格中的“动态修改”一列。此外, 在更新某个子任务时仅需提供需要更改的字段值, 无需提供完整的 Spec。

多媒体任务在创建后可由接口 POST mediatask/remove 去销毁, 而整个任务系列则由接口 DELETE mediatask/{MCS_Name}销毁。任务系列被销毁时, 所有仍在运行的任务也将同时被销毁。当一个任务系列被销毁后, 再次调用 POST mediatask/create_update 传递包含相同根“Name”字段的 MCS 将创建一个新的任务系列, 与之前的同名任务系列没有关系, 也不能引用其中的 Spec。

参数说明

路径

无

Header

Token:{token}

Body

MCS 结构。MCS 是一个多层次的 JSON 结构, 为方便描述, 下面采用逐层递进的方式描述。MCS 的一级结构如下:

字段名	说明
Name	必须, 字符串, 该 Spec 的名字。需保证唯一性, 后续会被其他结构/接口引用。

Type	必须，字符串，标识 Spec 的类型，即“MCS”。
Description	可选，字符串，该 Spec 的简短文字描述。
Version	可选，字符串，MCS 结构的版本信息，如“0.3”。
AudioSpecs	可选，结构数组，描述从音频采集设备生成音频流的参数。
VideoSpecs	可选，结构数组，描述从视频采集设备生成视频流的参数。
CodecStreamSpecs	可选，结构数组，描述由编码比特流生成音视频流的参数。
AudioAiSpecs	可选，结构数组，描述由音频流经 AI 处理生成新的数据流的参数。
VideoAiSpecs	可选，结构数组，描述由视频流经 AI 处理生成新的数据流的参数。
VideoDirectorSpec	可选，多级结构，描述多个视频流经导播后生成单路视频流的参数。
AudioComposeSpecs	可选，结构数组，描述多个音频流合成为一路音频流的参数。
VideoComposeSpecs	可选，结构数组，描述多个视频流合成为一路视频流的参数。
PlaySpecs	可选，结构数组，描述音频流播放的参数。
RenderSpecs	可选，结构数组，描述视频流显示的参数。
AudioCodecSpecs	可选，结构数组，描述音频流编码后生成音频编码比特流的参数。
VideoCodecSpecs	可选，结构数组，描述视频流编码后生成视频编码比特流的参数。
RecordSpecs	可选，结构数组，描述经由编码比特流生成录制文件的参数。
ServerSpecs	可选，结构数组，描述将编码比特流进行推流的参数。

表 22 MCS 一级结构

6.1.1 AudioSpecs 说明

AudioSpecs 用于描述通过本地音频采集/输入设备（麦克风）生成 PCM 原始数字音频流的参数配置。AudioSpecs 是一个 JSON 结构数组，每一个元素描述一条音频流生成的细节信息。每个元素的 JSON 结构构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 AudioSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
DeviceId	否	可选，字符串，音频输入设备 ID。不指定时使用本地 MIOS 中保存的默认音频输入设备。
SampleRate	否	可选，整数，音频输入设备采样率。单位为 Hz。默认使用本地 MIOS 中保存的该音频输入设备的 SampleRate 设置。
Channels	否	可选，整数，音频输入设备声道数。默认使用本地 MIOS 中保存的该音频输入设备的 Channels 设置。
Format	否	可选，字符串，音频输入设备采样的数据格式，如“S16LE”。默认使用本地 MIOS 中保存的该音频输入设备的 Format 设置。
Layout	否	可选，字符串，音频采样帧不同 channel 的数据样本的组合方式，如“Interleaved”。默认使用本地 MIOS 中保存的该音频输入设备的 Layout 设置。

表 23 AudioSpecs 结构

6.1.2 VideoSpecs 说明

VideoSpecs 用于描述通过本地视频采集/输入设备（摄像头）生成 YUV 原始数字视频流的参数配置。VideoSpecs 是一个 JSON 结构数组，每一个元素描述一条视频流生成的细节信息。对应的 JSON 结构的构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 VideoSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
DeviceId	否	可选，字符串，视频输入设备 ID。不指定时使用本地 MIOS 中保存的默认视频输入设备。
Width	否	可选，整数，视频输入分辨率的宽度。默认使用本地 MIOS 中保存的该视频输入设备的 Width 设置。
Height	否	可选，整数，视频输入分辨率的高度。默认使用本地 MIOS 中保存的该视频输入设备的 Height 设置。
FrameRate	否	可选，整数，视频帧率。默认使用本地 MIOS 中保存的该视频输入设备的 FrameRate 设置。
Format	否	可选，字符串，视频采集格式。如“mjpeg”，“YUY2”，“h.264”等。默认使用本地 MIOS 中保存的该视频输入设备的 Format 设置。

表 24 VideoSpecs 结构

6.1.3 CodecStreamSpecs 说明

CodecStreamSpecs 用于描述通过编码比特流（包括媒体文件或实时多媒体流）解码生成 PCM 数字音频流，YUV 数字视频流的细节。CodecStreamSpecs 是一个 JSON 结构数组，每一个元素均是一个 JSON 结构，描述一条多媒体流（可能同时含有音频流和视频流）生成的细节信息。该 JSON 结构的构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 CodecStreamSpec 的名字。需保证唯一性，可能会被其他 JSON 结构所引用。
URI	否	必须，字符串，标识编码比特流的地址，以 URI 的方式给出。注：比特流的协议格式需要包含在 URI 中，如 file://标识一个文件流协议，rtsp://标识一个 RTSP 协议的比特流。
AudioCodec	否	可选，字符串，标识音频部分的编码格式。如“aac”，“opus”。如不提供该字段则表示不包含音频流。
SampleRate	否	可选，整数。原始音频流的采样率，单位为 Hz。对于音频编码比特流，若流中不包含音频结构信息（即音频采样率，声道数）时，则必须提供该参数。
Channels	否	可选，整数。原始音频流的声道数。对于音频编码比特流，若流中不包含音频结构信息（即音频采样率，声道数）时，则必须提供该参数。

VideoCodec	否	可选，字符串，标识视频部分的编码格式。如“h.264”，“h.265”。如不提供该字段则表示不包含视频流。
Width	否	可选，整数，编码视频流的宽度。对于视频编码比特流，若流中不包含视频结构信息（即视频分辨率及帧率），则必须提供该参数。
Height	否	可选，整数，编码视频流的高度。对于视频编码比特流流，若流中不包含视频结构信息（即视频分辨率及帧率），则必须提供该参数。
FrameRate	否	可选，整数，编码视频流的帧率。对于视频编码比特流流，若流中不包含视频结构信息（即视频分辨率及帧率），则必须提供该参数。

表 25 CodecStreamSpecs 结构

6.1.4 AudioAiSpecs 说明

AudioAiSpecs 用于描述对音频流进行 AI 处理生成新的数据流的细节。AudioAiSpecs 是一个 JSON 数组，每个元素描述一个音频流进行 AI 处理生成新的数据流的相关参数。对应的 JSON 结构的构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 AudioAiSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
SourceName	否	必须，字符串，描述需要进行 AI 处理的音频流信息。为某一音频流 Spec 的“Name”字段设定。
Algorithm	否	可选，字符串，AI 处理算法名。如不指定时，并且“SourceName”所引用的音频流为本地某一音频输入设备时，使用该设备在本地 MIOS 中保存的默认“Algorithm”设置。其他情况必须指定该字段。

表 26 AudioAiSpecs 结构

6.1.5 VideoAiSpecs 说明

VideoAiSpecs 用于描述对视频流进行 AI 处理生成新的数据流的细节。VideoAiSpecs 是一个 JSON 数组，每个元素为一个 JSON 结构，表示一个视频流 AI 处理参数集。该 JSON 结构的构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 VideoAiSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
SourceName	否	必须，字符串，描述需要进行 AI 处理的视频流信息。为某一视频流 Spec 的“Name”字段设定。
Algorithm	否	可选，字符串，AI 算法名。如不指定时，并且“SourceName”所引用的视频流为本地某一视频输入设备时，使用该设备在本地 MIOS 中保存的默认“Algorithm”设置。其他情况必须指定该字

		段。
Width	否	可选，整数，算法输出视频分辨率的宽度。默认与视频输入分辨率一致。
Height	否	可选，整数，算法输出视频分辨率的高度。默认与视频输入分辨率一致。
FrameRate	否	可选，整数，算法处理输出视频的帧率。不可高于输入视频帧率，默认与其一致。

表 27 VideoAiSpecs 结构

6.1.6 VideoDirectorSpec 说明

VideoDirectorSpec 用于描述导播视频流生成的细节配置。该 JSON 结构的构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 VideoDirectorSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
Width	否	可选，整数，导播后形成视频流分辨率的宽度。默认使用本地 MIOS 中保存的导播流 Width 设置。
Height	否	可选，整数，导播后形成视频流分辨率的高度。默认使用本地 MIOS 中保存的导播流 Height 设置。
FrameRate	否	可选，整数，导播后形成视频流的帧率。设定不应大于导播输入任意一路视频流的帧率。默认使用本地 MIOS 中保存的导播流 FrameRate 设置。
VideoSourceSet	否	可选，字符串数组。指示导播流由哪些输入视频流生成。每一项都是某一视频流 Spec 中 Name 字段信息。当不提供 FrameSpec 时无需指定该字段，否则必须提供该字段。
FrameSpec	是	可选，结构，导播画面生成的细节参数。当不提供该结构时 MA 将根据当前已创建的 VideoAiSpec 集合，使用本地 MIOS 中保存的缺省导播策略模板。
Template	是	必须，字符串，产生导播流的分屏模板，只能从已有的模板中选择，当前仅有一个选项，即一分屏：“single frame”。
LayoutSpecs	是	必须，结构数组，描述分屏画面生成的细节参数。每一个元素描述某一分屏的画面生成参数。
Position	是	必须，整数，指示当前分屏位于整个画面的位置，必须是当前分屏模式接受的一个值。一分屏只接受 1。
SelectionSpecs	是	必须，结构数组，指示当前分屏画面的生成规则。每个元素描述一种状态下导播输出的视频流信息。
State	是	必须，字符串，状态名称。状态是 AI 算法对于画面内容理解的一种输出形式。例如“Stand_1”表示需要导播的视频流中出现一名学生处于站立状态。保留字“Others”用于指代没有任何没有明确描述的状态。
Priority	是	必须，整数，指示本条规则在整个导播策略集中的优先

		级。数值越小优先级越高。
VideoSource	是	必须，字符串，指示当前状态需要导播选择的输入视频流信息。必须是 VideoSourceSet 数组中的某一元素。
Delay	是	可选，整数，指示本条导播规则生效其状态需维持的最短时间，单位为秒。默认为 0。
Hold	是	可选，整数，指示本条导播规则生效后其视频维持的最短时间，单位为秒。默认为 0。
Timeout	是	可选，整数，指示本条导播规则生效后其视频维持的最长时间，单位为秒。超过该时限，即便当前规则的优先级仍为最高也将切换至下一优先级的导播规则。当 State 字段为“Others”时，该字段的内容将被忽略。默认没有最长时间限制。

表 28 DirectorSpec 结构

6.1.7 AudioComposeSpecs 说明

AudioComposeSpecs 用于描述合成音频流的参数配置。AudioComposeSpecs 是一个 JSON 结构数组，每个元素描述一条合成音频流的生成细节。对应的 JSON 结构构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 AudioComposeSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
SampleRate	否	可选，整数，合成音频流的采样率，单位为 hz。默认为 48000 hz。
MixerSpec	是	必须，结构数组，合成音频流的生成细节配置。
SourceName	是	可选，字符串，描述输入音频流信息。为某一音频流 Spec 的“Name”字段设定。
Volume	是	可选，整数，该输入音频流合成时的音量。若不指定则使用原始音频流的音量。

表 29 AudioComposeSpecs 结构

6.1.8 VideoComposeSpecs 说明

VideoComposeSpecs 用于描述合成视频流的参数配置。VideoComposeSpecs 是一个 JSON 结构数组，每个元素描述一条合成视频流的生成细节。对应的 JSON 结构构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 VideoComposeSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
Width	否	必须，整数，合成视频流分辨率的宽度。
Height	否	必须，整数，合成视频流分辨率的高度。
FrameRate	否	必须，整数，合成视频流帧率。注意：合成视频流帧率不能大于任一输入视频流帧率。
Background	是	可选，字符串，视频流的背景图片文件名，不指定时使用

		黑色背景。支持 jpeg, png 格式的文件。
CompositionSpec	是	必须，结构数组，合成视频流版面布局参数。
Geometry	是	可选，4 个元素构成的整数数组，用于描述合成视频流中的一个矩形的显示位置，以像素为单位。第一个元素是矩形框左上角在合成视频流上的水平坐标，第二个元素是矩形框左上角在合成视频流上的垂直坐标，第三个元素是矩形框的水平长度，第四个元素是矩形框的垂直高度。不指定该参数时表示合成视频流的完整矩形框。
ZOrder	是	可选，整数，取值范围 0-2，描述该矩形框与其他矩形框出现重叠时相对位置信息，默认为 0。数值越大的矩形框更靠近前景，其视频流/图片将显示在重叠区域。两个或多个 ZOrder 数值相同的矩形位置相互重叠，其行为未定义。
SourceName	是	可选，字符串，描述在该位置进行合成的输入视频流信息。为某一视频流 Spec 的“Name”字段设定。
Picture	是	可选，字符串，描述在该位置显示的图片文件。仅当“SourceName”字段不指定时使用。支持 jpeg, png 格式的文件。

表 30 VideoComposeSpecs 结构

6.1.9 VideoTransformSpecs 说明

VideoTransformSpecs 用于描述视频流经转换后形成新的视频流的参数配置。VideoTransformSpecs 是一个 JSON 结构数组，每个元素描述一条新视频流的生成细节。对应的 JSON 结构构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 VideoComposeSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
SourceName	否	必须，字符串，需要转换的视频流名称。为视频流 Spec 中“Name”字段的值。
Operation	否	必须，字符串，标识转换的需要执行操作。当前支持的操作列表如下： “Scale”：表示对原视频流做缩放操作 “Crop”：表示对原视频流做剪裁操作 “Flip”：表示对原视频流做翻转操作 “Rotate”：表示对原视频流做旋转操作 “RateAdjust”：表示对原视频流做帧率调整操作（请注意，帧率仅可向下调整即降低帧率不可向上调整） 需要执行的操作可以是多个操作合并进行，每个操作按其执行的顺序用连接符“ ”分割。
Width	否	可选，整数，仅当 Operation 字段包含“Scale”时存在，描述生成视频流分辨率的宽度。
Height	否	可选，整数，仅当 Operation 字段包含“Scale”时存在，描述生成视频流分辨率的宽度。

FlipMethod	是	可选，字符串。仅当 Operation 字段包含“Flip”时存在，描述翻转操作的方式。“Horizontal”表示以中心点水平翻转，“Vertical”表示以中心点做垂直翻转。
Rotation	是	可选，整数，可能值为 90,180, 270。仅当 Operation 字段包含“Rotate”时存在，描述旋转操作按顺时针方向旋转的度数。
Geometry	是	可选，4 个元素构成的整数数组，仅当 Operation 字段包含“Crop”时存在，用于描述剪裁画面在原始视频流上的位置，以像素为单位。第一个元素是剪裁矩形框左上角在原始视频流上的水平坐标，第二个元素是剪裁矩形框左上角在原始视频流上的垂直坐标，第三个元素是剪裁矩形框的水平长度，第四个元素是剪裁矩形框的垂直高度。
FrameRate	是	可选，整数，仅当 Operation 字段包含“RateAdjust”时存在，描述生成视频流的帧率。

表 31 VideoComposeSpecs 结构

6.1.10 PlaySpecs 说明

RenderSpecs 用于描述音频流播放的细节。PlaySpecs 是一个 JSON 结构数组，每个元素表示在一个音频播放设备上播放音频流的参数配置。对应的 JSON 结构构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 PlaySpecs 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
DeviceId	否	可选，字符串，播放设备 Id。如不指定，则使用本地 MIOS 中保存的默认音频输出设备播放。
Sources	是	必须，结构数组，播放设备输出音频流配置参数。注意：1. 每次修改需要提供需要播放的音频流完整描述。意味着不再出现的音频流将不被播放。2. 每路音频流将被独立播放而不会混音。如需混音请在播放前要使用 AudioComposeSpec。
SourceName	是	必须，字符串，描述需要输出的音频流名称。为某一音频流 Spec 中的“Name”字段值。
Volume	是	可选，整数，该路音频流输出时的音量配置。

表 32 PlaySpecs 结构

6.1.11 RenderSpecs 说明

RenderSpecs 用于描述视频流渲染的细节。RenderSpecs 是一个 JSON 结构数组，每个元素为一个 JSON 结构，表示在一个视频输出设备（显示器）上渲染视频流的参数配置。该 JSON 结构的构成如下：

字段名	动态修改	说明
-----	------	----

Name	否	必须，字符串，该 RenderSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
DeviceId	否	可选，字符串，回显设备 Id。如不指定，则使用本地 MIOS 中保存的默认视频输出设备播放。
Background	是	可选，字符串，视频流的背景图片文件名，不指定时使用黑色背景。支持 jpeg, png 格式的文件。
CompositionSpec	否	必须，结构数组，显示设备输出画面构成参数。注意：结构数组的大小不可动态修改，但每一个元素的设置可以修改。
Geometry	是	可选，4 个元素构成的整数数组，用于描述视频流在显示设备上的回显位置，以像素为单位。第一个元素是矩形框左上角在显示设备上的水平坐标，第二个元素是矩形框左上角在显示设备上的垂直坐标，第三个元素是矩形框的水平长度，第四个元素是矩形框的垂直高度。不指定该参数时表示合成视频流将占据整个屏幕。
ZOrder	是	可选，整数，取值范围 0-2，描述该矩形位置出现重叠时相对位置信息，默认为 0。数值越大的矩形位置月靠近前景，其视频流/图片将显示在重叠区域。两个或多个 ZOrder 数值相同的矩形位置相互重叠，其行为未定义。
SourceName	是	可选，字符串，在当前位置需要渲染的视频流名称。为视频流 Spec 的“Name”字段值。
Picture	是	可选，字符串，描述在该位置显示的图片文件。仅当“SourceName”字段不指定时使用。支持 jpeg, png 格式的文件。

表 33 RenderSpecs 结构

6.1.12 AudioCodecSpecs 说明

AudioCodecSpecs 用于描述音频流编码的细节。AudioCodecSpecs 是一个 JSON 结构数组，每个元素为一个 JSON 结构，表示一个音频编码参数集。该 JSON 结构的构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 AudioCodecSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
SourceName	否	必须，字符串，需要编码的音频流名称。为音频流 Spec 中“Name”字段的值。
Codec	否	可选，字符串，音频编码格式，当前仅支持“pcm”，“aac”，“opus”。不指定时使用本地 MIOS 中保存的默认音频编码格式。请注意某些编码格式对于输入的音频流格式有要求，如 opus 仅支持 8K, 12K, 16K, 24K, 48K 这几种采样率。调用方需保证“SourceName”引用的音频流符合编码格式的要求。
BitRateMode	是	可选，字符串，音频编码的比特率控制，“VBR”或“CBR”，默认使用本地 MIOS 中保存的该编码格式的 BitRateMode 设置。注：“pcm”，“aac”编码格式不支持该设置。
BitRate	是	可选，字符串，音频编码的比特率，结尾可为“bps”，“kbps”

		或“mbps”。默认使用本地 MIOS 中保存的该编码格式的 BitRate 设置。注：“pcm”编码格式不支持该设置。
--	--	--------------------------------------------------------------

表 34 AudioCodecSpecs 结构

6.1.13 VideoCodecSpecs 说明

VideoCodecSpecs 用于描述视频流编码的细节。VideoCodecSpecs 是一个 JSON 结构数组，每个元素为一个 JSON 结构，表示一个视频编码参数集。该 JSON 结构的构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 VideoCodecSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
SourceName	否	必须，字符串，需要编码的视频流名称。为视频流 Spec 中“Name”字段的值。
Codec	否	必须，字符串，视频编码格式，当前支持“h.264”，“h.265”，“vp9”。不指定时使用本地 MIOS 中保存的默认视频编码格式。
BitRateMode	是	可选，字符串，视频编码的比特率控制，“VBR”或“CBR”，默认为“VBR”，默认使用本地 MIOS 中保存的该视频编码格式的 BitRateMode 设置。
BitRate	是	可选，字符串，视频编码的比特率，后缀可为“bps”，“kbps”或“mbps”，默认使用本地 MIOS 中保存的该视频编码格式的 BitRate 设置。
iFrameInterval	是	可选，整数，视频编码 I 帧间隔。默认使用本地 MIOS 中保存的该视频编码格式的 iFrameInterval 设置。
idrInterval	是	可选，整数，视频编码 IDR 帧间隔。默认使用本地 MIOS 中保存的该视频编码格式的 idrInterval 设置。
QpRange	是	可选，两个元素的整数数组。用来表示 Qp 参数的取值范围。默认使用本地 MIOS 中保存的该视频编码格式的 QpRange 配置。
Profile	是	可选，字符串，视频编码 profile 名称。默认使用本地 MIOS 中保存的该视频编码格式的 Profile 配置。

表 35 VideoCodecSpecs 结构

6.1.14 RecordSpecs 说明

RecordSpecs 用于描述通过编码比特流录制文件的细节。RecordSpecs 是一个 JSON 结构数组，包含多个相同结构 JSON 元素，每个元素表示一组录制文件的参数集。该 JSON 结构的构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 RecordSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
AudioCodecName	否	可选，字符串，描述引用的音频编码比特流。即

		AudioCodecSpec 中“Name”字段的值。
VideoCodecName	否	可选，字符串，描述引用的视频编码比特流。即 VideoCodecSpec 中“Name”字段的值。
Format	否	可选，字符串，录制文件容器格式，默认为“mp4”。支持的容器格式为：视频及音视频：“mp4”，“mkv”，“flv”，音频：“ogg”，“aac”，“wav”。
Path	否	必须，字符串，录制文件名（含路径），名字中支持包含时间戳模板，如 yyyy-mm-dd-hh-mm-ss.mp4。注意：文件名中可以包含用“/”分割的相对路径信息，用以表示在 Media Agent 目标存储位置下的目录信息。
SplitMethod	否	可选，字符串，指示是否需要分段录制的分段方法。仅在需要分段录制时提供该字段。当前支持两种分段模式“Time”和“Size”，分别表示按时间和按文件大小分段。
SplitUnit	是	可选，整数，指示分段的大小。仅当 SplitMethod 存在时提供。当 SplitMethod 为“Time”时，该数值的单位为秒。当 SplitMethod 为“Size”时，该数值的单位为字节。

表 36 RecordSpecs 结构

6.1.15 ServerSpecs 说明

ServerSpecs 用于描述通过编码比特流进行推流细节。ServerSpecs 是一个 JSON 结构数组，每个元素描述一组推流参数集。该 JSON 结构的构成如下：

字段名	动态修改	说明
Name	否	必须，字符串，该 ServerSpec 的名字。需保证唯一性，后续可能会被其他 JSON 结构引用。
AudioCodecName	否	可选，字符串，描述引用的音频编码比特流。即 AudioCodecSpec 中“Name”字段的值。
VideoCodecName	否	可选，字符串，描述引用的视频编码比特流。即 VideoCodecSpec 中“Name”字段的值。
URI	否	必须，字符串，视频推流目标地址。注意 URI 中应标识目标协议，如“rtsp://”标识采用 RTSP 协议推流。

表 37 ServerSpecs 结构

响应说明

Header

可能包含更新的 Token 键值。

Body

JSON 结构，除公共字段外的结构如下：

字段名	说明
Results	可选，结构数组，子任务执行的结果。
SpecName	必须，字符串，某一子任务对应的 Spec 名。为 Spec 中“Name”字段的值。
Success	必须，布尔值，标识任务是否启动成功。

Reason	可选，字符串，当“Success”为 false 时描述任务失败的原因。
RecordPath	可选，字符串，录制子任务启动成功后最终录制文件路径信息。

表 38 POST mediatask/create 接口响应 Body 结构

示例

请求


POST mediatask/create_update

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body


MCS_Example_E
xt.json

响应

Header

无

Body

```
{
  "Code" : 201,
  "Message" : "Task created successfully!",
  "Results" : [
    {
      "SpecName" : "Default_Audio",
      "Success" : true
    },
    {
      "SpecName" : "Teacher",
      "Success" : true
    },
    {
      "SpecName" : "Student",
      "Success" : true
    },
    {
      "SpecName" : "Student_20",
      "Success" : true
    },
    {
      "SpecName" : "Student_30",
      "Success" : true
    },
    {
      "SpecName" : "Teacher_Track",
      "Success" : true
    },
    {

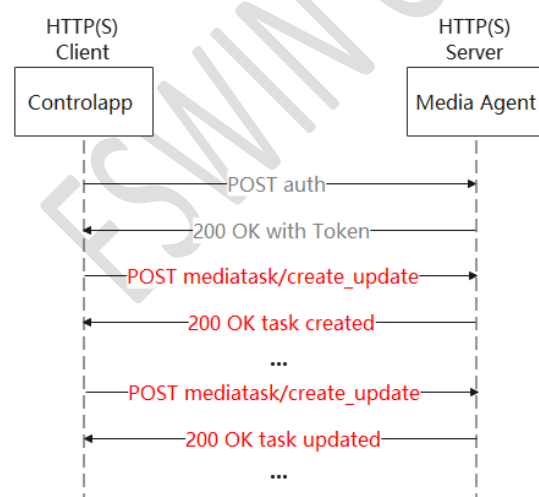
```

```

    "SpecName" : "Student_Track",
    "Success" : true
  },
  {
    "SpecName" : "Director",
    "Success" : true
  }
  {
    "SpecName" : "Default_Speaker",
    "Success" : true
  },
  {
    "SpecName" : "Main_Display",
    "Success" : true
  },
  {
    "SpecName" : "Default_Audio_AAC",
    "Success" : true
  }
  {
    "SpecName" : "Director_264",
    "Success" : true,
    "RecordPath" : "director_2022-03-01-08-00-00.mp4"
  }
  {
    "SpecName" : "Director_Record",
    "Success" : true
  },
  {
    "SpecName" : "Director_RTSP",
    "Success" : true
  }
}
]
}

```

时序



6.1.16 MIOS

MIOS 用于和 MCS 配合，提供 MCS 应用过程中使用设备，算法，编解码相关的所缺省参数。

MIOS 由本地管理系统负责创建、更新，修改需经由本地管理系统进行。通过 mediadevice 类接口，用户可以获得当前系统 MIOS 的配置细节。MIOS 是一个 JSON 结构，其构成如下：

字段名	说明
AudioEnCodes	可选，结构数组，支持音频编码算法及默认参数列表。
Default	可选，布尔值，标识该音频编码格式是否为默认的音频编码格式。仅有一个元素该字段为 true，其余元素应为 false。默认设置为 false。
Codec	必须，字符串，音频编码格式，如“opus”。
BitRateMode	必须，字符串，该音频编码的比特率控制，“VBR”或“CBR”。
BitRate	必须，字符串，该音频编码的比特率。
AudioDeCodes	可选，结构数组，支持音频解码算法及默认参数列表。
Codec	必须，字符串，音频编码格式，如“opus”。
VideoEnCodes	可选，结构数组，支持视频编码算法及默认参数列表。
Default	可选，布尔值，标识该视频编码格式是否为默认的视频编码格式。仅有一个元素该字段为 true，其余元素应为 false。默认设置为 false。
Codec	必须，字符串，视频编码格式，如“h.264”，“h.265”，“vp8”，“vp9”。
BitRateMode	必须，字符串，视频编码的比特率控制，“VBR”或“CBR”。
BitRate	必须，字符串，视频编码的比特率，后缀可为“bps”，“kbps”或“mbps”。
iFrameInterval	必须，整数，视频编码 I 帧间隔。
idrInterval	必须，整数，视频编码 IDR 帧间隔。
QpRange	必须，两个元素的整数数组。用来表示 Qp 参数的取值范围。
Profile	必须，字符串，视频编码 profile 名称，默认为“main”。
VideoDeCodes	可选，结构数组，支持的视频解码算法及默认参数列表。
Codec	必须，字符串，视频解码格式，如“h.264”，“h.265”，“vp8”，“vp9”。
Audiolns	可选，结构数组，音频输入设备列表。
Enable	可选，布尔值，标识该设备是否可用。缺省设置 true。
Connected	必须，布尔值，标识该设备是否已连接到 Metis Station。
Default	可选，布尔值，标识该设备是否作为缺省音频输入设备。仅有一个元素该字段为 true，其余元素均应为 false。缺省设置 false。
DeviceName	必须，字符串，音频输入设备的系统名。
DeviceId	必须，字符串，音频输入设备的系统 ID。
AliasName	可选，字符串，音频输入设备的别名。
SubDevices	可选，结构数组，为该音频输入设备下的子设备列表。
Enable	可选，布尔值，子设备是否可用。缺省设置 true。
Connected	必须，布尔值，子设备是否已连接到 Metis Station。
Id	必须，字符串，子设备 ID。
Name	必须，字符串，子设备名。
Alias	可选，字符串，子设备别名。
Type	必须，字符串，音频输入设备类型，当前只有一种可能“MIC”。
CanLocalAmplification	可选，布尔值，音频输入设备是否支持本地扩声。
LocalAmplification	可选，布尔值，音频输入设备是否使能本地扩声。

SampleRate	必须，整数，音频输入设备采样率。单位为 hz。
Channels	必须，整数，音频输入设备采样时的声道配置。
Format	必须，字符串，音频输入设备采样的数据格式，如“S16LE”
Layout	必须，字符串，音频采样帧不同 channel 的数据样本的组合方式，如“Interleaved”。
Algorithms	可选，字符串，为该设备配置的 AI 处理算法。
Name	必须，字符串，算法名。
Default	可选，布尔值，标识算法是否是该设备的默认处理算法。仅有一个元素该字段为 true，其余元素均应为 false。缺省设置 false。
Format	必须，字符串，AI 处理后的数据流格式。如“Audio”，“Data”，“Audio+Data”。
AudioOuts	可选，结构数组，音频输出设备列表。
Enable	可选，布尔值，标识该设备是否可用。缺省设置 true。
Connected	必须，布尔值，标识该设备是否已连接到 Metis Station。
Default	可选，布尔值，标识该设备是否作为缺省音频输出设备。仅有一个元素该字段为 true，其余元素均应为 false。缺省设置 false。
DeviceName	必须，字符串，音频输出设备的系统名。
DeviceId	必须，字符串，音频输出设备的系统 ID。
AliasName	可选，字符串，音频输出设备的别名。
SubDevices	可选，结构数组，为该音频输出设备下的子设备列表。
Enable	可选，布尔值，子设备是否可用。缺省设置 true。
Connected	必须，布尔值，子设备是否已连接到 Metis Station。
Id	必须，字符串，子设备 ID。
Name	必须，字符串，子设备名。
Alias	可选，字符串，子设备别名。
VideoIns	可选，结构数组，视频输入设备列表。
Enable	可选，布尔值，标识该设备是否可用。缺省设置 false。
Connected	必须，布尔值，标识该设备是否已连接到 Metis Station。
Default	可选，布尔值，标识该设备是否作为缺省视频输入设备。仅有一个元素该字段为 true，其余元素均应为 false。缺省设置 false。
DeviceName	必须，字符串，视频输入设备的系统名。
DeviceId	必须，字符串，视频输入设备的系统 ID。
AliasName	可选，字符串，视频输入设备的别名。
Type	必须，字符串，视频输入设备类型，如“USB”，“HDMI”，“WEB”。
Format	必须，字符串，默认使用的视频采集格式。如“mjpeg”，“YUY2”，“h.264”等。
URI	可选，字符串，仅在 Type 为“WEB”时使用，用于标识网络摄像头产生的网路流的源地址。
Width	必须，整数，默认使用的采集分辨率宽度。
Height	必须，整数，默认使用的采集分辨率高度。
FrameRate	必须，整数，默认使用的采集视频帧率。
Algorithms	可选，字符串，为该设备配置的 AI 处理算法。
Name	必须，字符串，算法名。

Default	可选，布尔值，标识算法是否是该设备的默认处理算法。仅有一个元素该字段为 true，其余元素均应为 false。缺省设置 false。
Format	必须，字符串，AI 处理后的数据流格式。如“Video”，“Data”，“Video+Data”。
PresetPositions	可选，结构数组，仅当 Algorithm 不为空时使用。算法需要的预置位配置。
Geometry	必须，4 个元素构成的浮点数数组，用于描述算法需要使用的一个预置位的位置信息。
Type	必须，整数，预置位类型。
Tag	必须，字符串，预置位描述。
VideoOuts	可选，结构数组，视频输出设备列表。
Enable	可选，布尔值，标识该设备是否可用。缺省设置为 true。
Connected	必须，布尔值，标识该设备是否已连接到 Metis Station。
Default	可选，布尔值，标识该设备是否作为缺省视频输出设备。仅有一个元素该字段为 true，其余元素均应为 false。缺省设置 false。
DeviceName	必须，字符串，视频输出设备的系统名。
DeviceId	必须，字符串，视频输出设备的系统 ID。
AliasName	必须，字符串，视频输出设备的别名。
StartPosition	必须，2 个元素构成的整数数组，用于描述输出设备左上角的坐标值。
Width	必须，整数，视频输出设备分辨率宽度。
Height	必须，整数，视频输出设备分辨率高度。
VideoDirectorParas	可选，结构，导播流的配置信息。
Width	必须，整数，导播后形成视频流分辨率的宽度。
Height	必须，整数，导播后形成视频流分辨率的高度。
FrameRate	必须，整数，导播后形成视频流的帧率。
DefaultDirectorSpecs	可选，结构数组，不同场景下默认导播参数。
Scenario	必须，字符串，标识导播参数适合的场景。只能是“Teacher+Student+Ppt+Blackboard”或“Teacher+Student+Ppt”或“Teacher+Student”。
FrameSpec	必须，结构，导播画面生成的细节参数。
Template	必须，字符串，产生导播流的分屏模板，当前仅有一个选项，即一分屏：“TEMPLATE_ONE_FRAME”。
LayoutSpecs	必须，结构数组，描述分屏画面生成的细节参数。
Position	必须，整数，指示当前分屏位于整个画面的位置。
SelectionSpecs	必须，结构数组，指示当前分屏画面的生成规则。
State	必须，字符串，状态名称。
Priority	必须，整数，指示本条规则在整个导播策略集中的优先级。
VideoSource	必须，字符串，指示当前状态需要导播选择的输入视频流信息。必须是当前场景支持的视频流，例如“Teacher+Student”场景下该字段的设定只可以是“Teacher”，“Student”，“Teacher_Closeup”或“Student_Closeup”。
Delay	必须，整数，指示本条导播规则生效其状态需维持的最短时间，

	单位为秒。
Hold	必须，整数，指示本条导播规则生效后其视频维持的最短时间，单位为秒。
Timeout	必须，整数，指示本条导播规则生效后其视频维持的最长时间，单位为秒。当 State 字段为“Others”时，该字段的内容将被忽略。

表 39 ProcessSpecs 结构

6.2 POST mediatask/remove

功能说明

用于在销毁之前由 MCS 创建的多媒体任务。需要销毁的任务通过 MCS 中各 Spec 的“Name”字段标识。

参数说明

路径

无

Header

Token:{token}

Body

JSON 结构，详细描述如下：

字段名	说明
Name	必须，字符串，MCS 名称。为之前 POST mediatask/create_update 接口调用时传递的 MCS 中根“Name”字段的值。
SpecNames	必须，字符串数组，每一项均为某一子任务对应的 Spec 名，即为 Spec 中“Name”字段的值。

表 40 POST mediatask/remove 接口 Body 参数结构

响应说明

Header

可能包含更新的 Token 键值。

Body

除公共字段外没有其他字段。

示例

请求

POST mediatask/remove

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzO

TA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "Name" : "Example",
  "SpecNames": ["Director_RTSP"]
}
```

响应

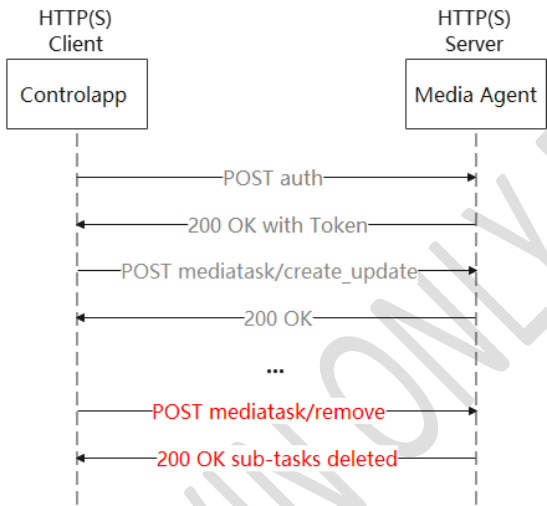
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
}
```

时序



6.3 POST mediatask/suspend

功能说明

用于暂停由 MCS 创建的多媒体任务的执行。请注意当前仅支持暂停录制子任务，即由 RecordSpec 创建出来的子任务。

参数说明

路径

无

Header

Token:{token}

Body

JSON 结构，详细描述如下：

字段名	说明
Name	必须，字符串，MCS 名称。为之前 POST mediatask/create_update 接口调用时传递的 MCS 中“Name”字段的值。
SpecNames	必须，字符串数组，每一项均为某一子任务对应的 Spec 名，即为 Spec 中“Name”字段的值。

表 41 POST mediatask/suspend 接口 Body 参数结构

响应说明

Header

可能包含更新的 Token 键值。

Body

除公共字段外没有其他字段。

示例

请求

POST mediatask/suspend

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "Name" : "Example",
  "SpecNames": ["Director_Record"]
}
```

响应

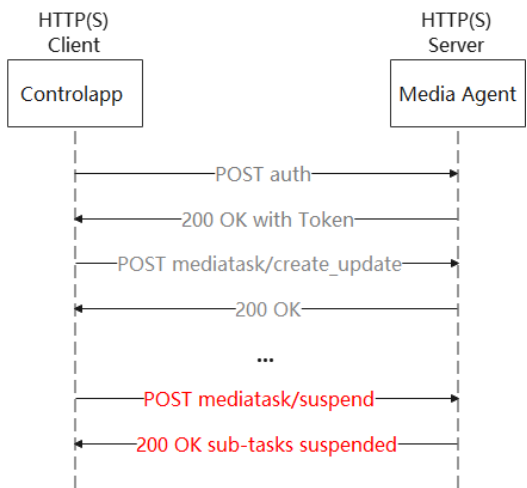
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
}
```

时序



6.4 POST mediatask/resume

功能说明
用于恢复之前暂停的多媒体任务的执行。请注意当前仅支持暂停，恢复录制任务，即由 RecordSpec 创建出来的任务。

参数说明

路径
无

Header
Token:{token}

Body
JSON 结构，详细描述如下：

字段名	说明
Name	必须，字符串，MCS 名称。为之前 POST mediatask/create_update 接口调用时传递的 MCS 中“Name”字段的值。
SpecNames	必须，字符串数组，每一项均为某一子任务对应的 Spec 名，即为 Spec 中“Name”字段的值。

表 42 POST mediatask/resume 接口 Body 参数结构

响应说明
Header
可能包含更新的 Token 键值。

Body
除公共字段外没有其他字段。

示例
请求
POST mediatask/resume

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "Name" : "Example",
  "SpecNames": ["Director_Record"]
}
```

响应

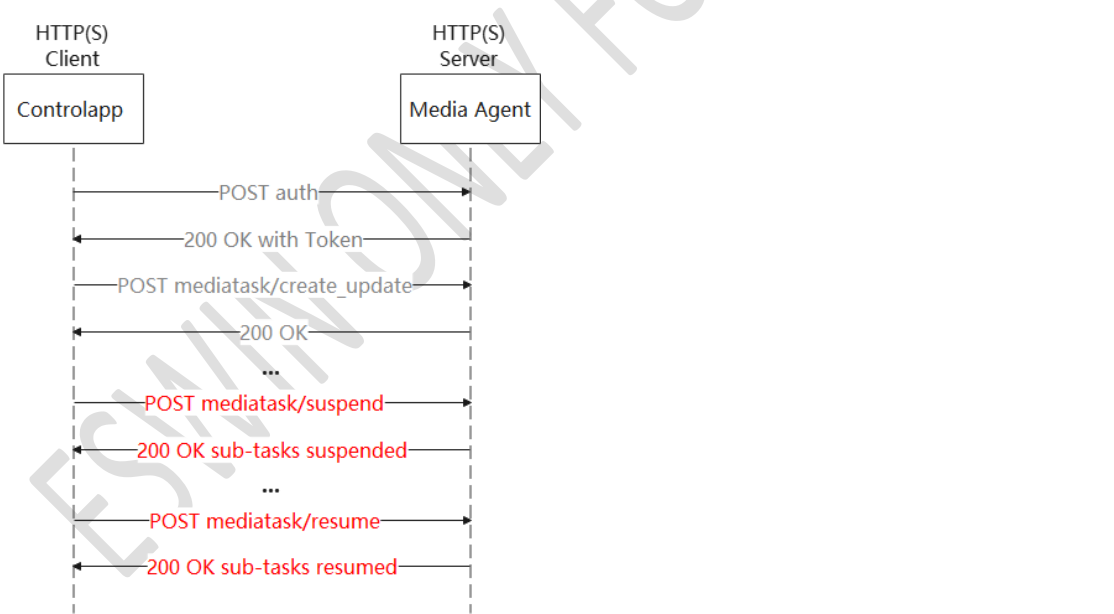
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
}
```

时序



6.5 POST mediatask/inspection

功能说明

用于查询某个任务系列的 MCS 完整描述。

参数说明

路径

无

Header

Token:{token}

Body

JSON 结构，详细描述如下：

字段名	说明
Name	必须，字符串，MCS 名称。为之前 POST mediatask/create 接口调用时传递的 MCS 中“Name”字段的值。

表 43 POST mediatask/inspection 接口 Body 参数结构

响应说明

Header

可能包含更新的 Token 键值。

Body

对应该任务系列的 MCS。

示例

请求

POST mediatask/inspection

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "Name" : "Example"
}
```

响应

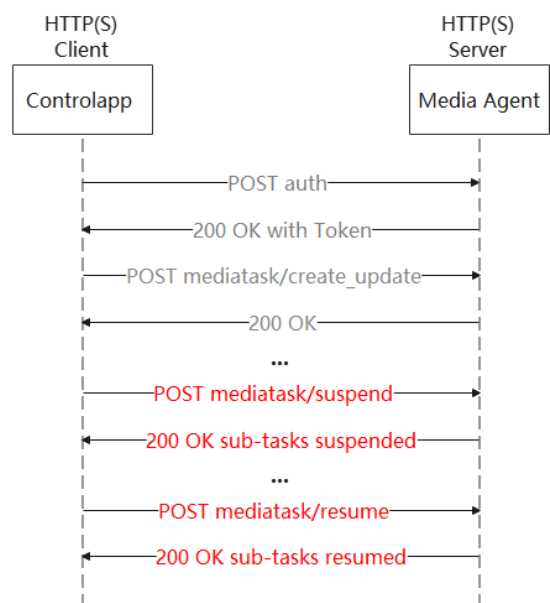
Header

无

Body

略。

时序



6.6 POST mediatask/keyframe

功能说明

用于触发视频编码子任务立即产生一个 IDR 帧。

参数说明

路径

无

Header

Token:{token}

Body

JSON 结构，详细描述如下：

字段名	说明
Name	必须，字符串，MCS 名称。为之前 POST mediatask/create_update 接口调用时传递的 MCS 中根“Name”字段的值。
SourceName	必须，字符串，视频编码子任务（由 VideoCodecSpec 创建）名称，即 VideoCodecSpec 中“Name”字段的值。

表 44 POST mediatask/keyframe 接口 Body 参数结构

响应说明

Header

可能包含更新的 Token 键值。

Body

除公共字段外没有其他字段。

示例
请求

POST mediatask/keyframe

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWVRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "Name" : "Example",
  "SourceName" : "Director_264"
}
```

响应

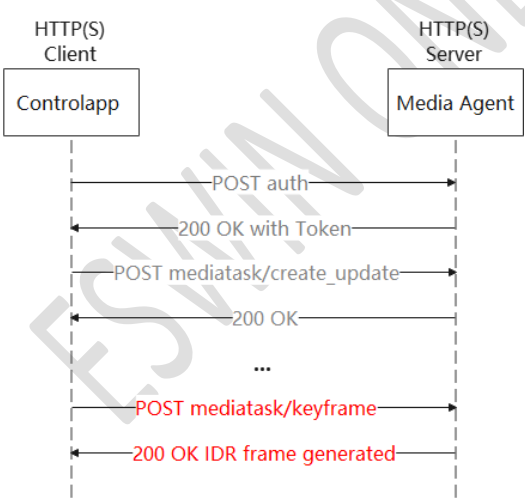
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
}
```

时序



6.7 POST mediatask/snapshot

功能说明

用于触发视频子任务立即产生一个快照。

参数说明

路径
无

Header
Token:{token}

Body
JSON 结构，详细描述如下：

字段名	说明
Name	必须，字符串，MCS 名称。为之前 POST mediatask/create_update 接口调用时传递的 MCS 中根“Name”字段的值。
SourceName	必须，字符串，需要获取快照的视频流名称。为视频流 Spec 中“Name”字段的值。
Format	必须，字符串，快照文件格式，如“png”。支持 png, jpeg 格式
Path	必须，字符串，录制文件名，支持时间戳模板，如 yyyy-mm-dd-hh-mm-ss.png。

表 45 POST mediatask/snapshot 接口 Body 参数结构

响应说明
Header
可能包含更新的 Token 键值。

Body
除公共字段外没有其他字段。

示例
请求
POST mediatask/snapshot

参数
Header
Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

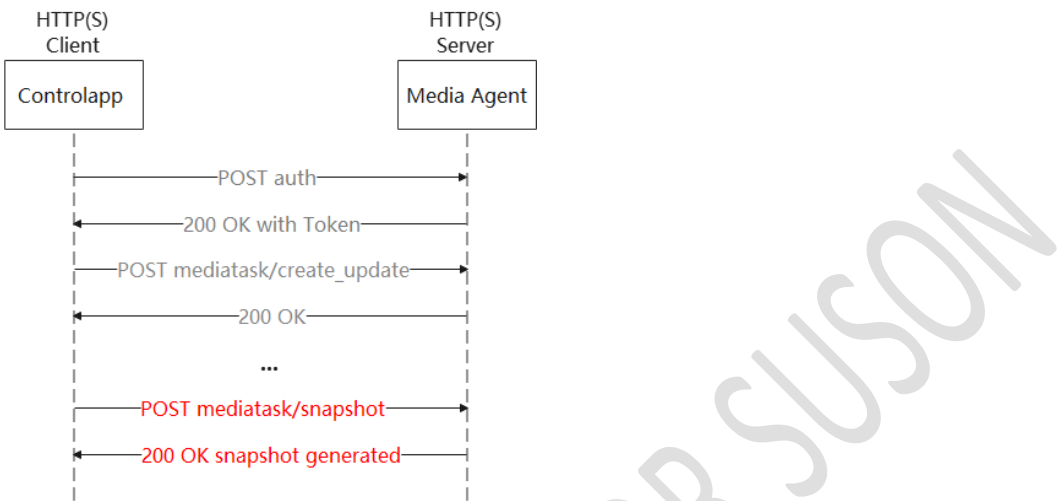
Body
{
 "Name" : "Example",
 "SourceName" : "Director",
 "Format" : "png",
 "Path" : "Director-yyyy-mm-dd-hh-mm-ss.png"
}

响应
Header
无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!"
}
```

时序



6.8 POST mediatask/merge

功能说明

用于将多个相同格式的多媒体文件合并成一个多媒体文件。

参数说明

路径

无

Header

Token:{token}

Body

JSON 结构，详细描述如下：

字段名	说明
Paths	必须，字符串数组，需要合并的文件名（含相对路径）列表。注意：文件合并过程按照原文件在数组中出现的次序进行，因此调用方需确保合并文件以正确的顺序出现在该数组中。支持合并的文件容器类型为 mp4, mkv, flv, ogg, aac, wav。
MergedFile	必须，字符串，录制文件名（含路径）注意：文件名中可以包含用"/"分割的相对路径信息，用以表示在 Media Agent 目标存储位置下的目录信息。仅当 KeepSource 设定为 false 时，该字段的设定可以与 Paths 列表中的某一项一致。

KeepSource	可选，布尔值，是否在合并完成后删除源文件，默认为 false，即不再保留。如果合并失败，不论该字段设定为何，均不会删除源文件。
------------	-----------------------------------------------------------------

表 46 POST mediatask/merge 接口 Body 参数结构

响应说明

Header

可能包含更新的 Token 键值。

Body

除公共字段外没有其他字段。

示例

请求

POST mediatask/snapshot

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJiMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

```
{
  "Paths": ["Director_01.mp4", "Director_02.mp4", "Director_03.mp4"]
}
```

响应

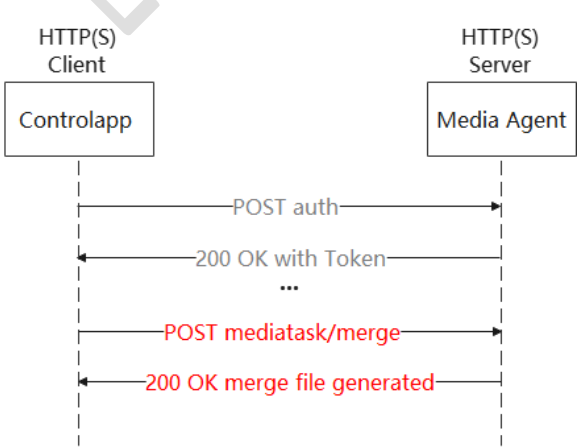
Header

无

Body

```
{
  "Code": 200,
  "Message": "No error found!"
}
```

时序



6.9 DELETE mediatask/{MCS_Name}

功能说明

用于销毁之前由“Name”指示的由 MCS 创建的多媒体任务系列。

参数说明

路径

MCS_Name: 必须, 字符串, 需要停止的多媒体任务名。为之前 POST mediatask/create 接口调用时传递的 MCS “Name”字段值。

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值

Body

除公共字段外没有其他字段。

示例

请求

```
DELETE mediatask/Example
```

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

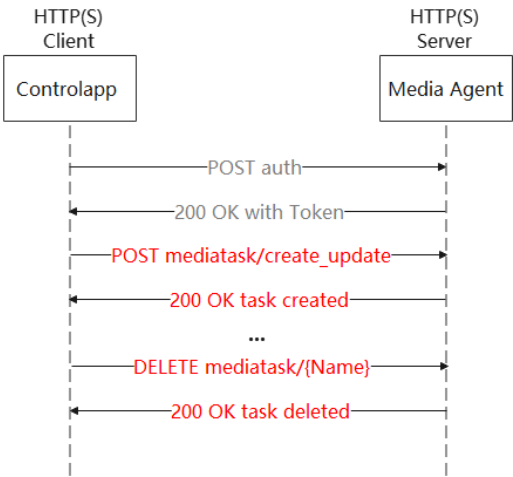
Header

无

Body

```
{  
  "Code" : 200,  
  "Message" : "No error found!",  
}
```

时序



6.10 GET mediatask/tasks

功能说明

用于获取用户当前运行的多媒体任务系列的列表。

参数说明

路径

无

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值

Body

JSON 结构，除公共字段外的结构如下：

字段名	说明
Names	必须，字符串数组，每一个元素为 MCS 根“Name”字段值。

表 47 GET mediatask/tasks 接口响应 Body 结构

示例

请求

GET mediatask/tasks

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

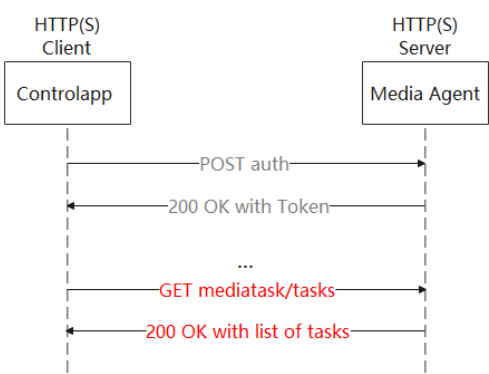
Header

无

Body

```
{
  "Code" : 200,
  "Message" : "No error found!",
  "Names" : ["Example"]
}
```

时序



6.11 GET mediatask/{MCS_Name}/{Spec_Name}

功能说明

用于获取指定多媒体任务的详细运行信息。

参数说明

路径

MCS_Name: MCS 根“Name”字段值，标识任务系列。

Spec_Name: 关心的 Spec “Name”字段值，标识任务系列中的一个任务。

Header

Token:{token}

Body

无

响应说明

Header

可能包含更新的 Token 键值

Body

JSON 结构，除公共字段外的结构如下：

字段名	说明
SpecType	必须，字符串，子任务类型。如 "AudioSpec"，"VideoSpec"，"CodecStreamSpec"等。
Duration	必须，字符串，子任务运行时间。格式为"xxhxxmxxs"
Path	可选，字符串数组，录制文件名（含路径）列表（分段录制可能会生成多个文件）。仅当 Type 为"Record"存在。
Layouts	可选，字符串数组，描述在视频输出设备各个位置渲染时的视频流名，即对应 Spec 的"Name"字段值。仅当 Type 为"RenderSpec"时存在。

表 48 GET mediatask/tasks 接口响应 Body 结构

示例

请求

GET mediatask/Example/Main_Display

参数

Header

Token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VySWQiOiJlMDhmODZhZi0zNWRRhLTQ4ZjltOGZhYi1jZWYzOTA0NjYwYmQifQ.-xN_h82PHVTCMA9vdoHrcZxH-x5mb11y1537t3rGzcM

Body

无

响应

Header

无

Body

```
{
  "Code": 200,
  "Message": "No error found!",
  "SpecType": "RenderSpec",
  "Duration": "00h30m:00s",
  "Layouts": [
    "Director",
    "Teacher",
    "Student_20",
    "Student_30"
  ]
}
```

时序

