Leading Solution Provider for IP-based Video Transmission

API

CUBE R1

NDI High Bandwidth+NDI|HX

Multi-Channel Recording System

(2023.11)

Through this manual, you will learn:

- Interface descriptions of the API
- Methods for using the API

As the product undergoes continuous upgrades and updates, the content of this manual may vary from the product you have purchased. Please refer to the actual product within the package for accurate information.

Legal Statement

By receiving this document from Changsha Kiloview Electronic Technology Co., Ltd. (referred to as "Kiloview" hereafter), you agree to the following terms. If you do not agree with these terms, please discontinue the use of this document.

This document is copyrighted by Changsha Kiloview Electronic Technology Co., Ltd. All rights not expressly granted in this document are reserved by Kiloview. This document contains proprietary information of Kiloview. Without prior written consent from Kiloview, no individual or entity shall copy, transmit, distribute, use, or disclose this document, including any images, tables, data, and other information contained herein.

KILOVIEW® 千视 is a registered trademark of Kiloview. Names and logos of Kiloview products are trademarks or registered trademarks of Kiloview. Other product or company names mentioned in this document may be trademarks or registered trademarks of their respective owners. Reading this document does not imply, by any means, an assignment of rights to the reader with respect to any mark appearing in this document, unless prior written consent is obtained from Kiloview or the respective third-party rights holder.

This product complies with design requirements related to environmental protection and personal safety. Storage, use, and disposal of the product should adhere to the requirements of the product manual, relevant contracts, and the laws and regulations of the respective country.

This document is provided on an "as is" and "as available" basis. Information in this document is subject to change as Kiloview products and technologies evolve, and Kiloview is not obligated to inform about such updates.

For matters not covered in this document, please visit Kiloview's website at www.kiloview.com for relevant information and technical support.

CONTENT

Welcome

Cube R1 HTTP API is the programming interface for Cube R1 network decoding and software development.Aimed at simple HTTP request/responding mechanism, NDI R1 HTTP API wants to realize various functions of controlling for device. Software developers make use of NDI R1 HTTP API to meet the purpose of remote management and device control.

Each function of API corresponds to one HTTP request path. In current version, the path format is http://[:]/api/[module name]/[function].json

This file will list each API's corresponding filename and functions descriptions. Meanwhile, in each API description file, there will have one testing interface. You could write your request testing parameter and execute test according to parameter description. It is very convenient to testify API function online.

API requests follow the standard HTTP protocol. Developers can use any tools, software development library that according with HTTP standard (For example Web browser, JavaScript library, C/C++ HTTP protocol file, JAVA, .NET etc.) Submit of parameters might be POST or GET, or both (please refer to API instruction for details) according to different API interfaces request. API's

response is JSON format. Please refer to each API instruction. We’ ve tried our best to maintain the conformity of API interfaces, but there may some differences due to products difference leading different API interfaces. Therefore, developers are requested to notice the difference according to the guidance.

Instruction for calling interface

The return forms of all interfaces are unified as：

Normal return

```
{
    "result":"ok",
    "data": {} // " Certain types of data, such as strings, numbers, arrays,
objects, etc., may not have this field "
}
```

Error Return

```
{
    "result":"error",
    "msg":" Error: Invalid request method!",
    "reason":"api:method"
}
```

> Note: When msg is a numerical value, you need to call the MsgCode interface to get thecontent of the readable error message, for example: 1100

Example:

```
{
    "1100":{
        "en":"Unabletoconnecttosystemservices",
        "zh":"无法连接系统服务"
    }
}
```

Module list

Module name	Instructions
config	Software configurations
user	User management
source	Source Input
layout	Layout Settings
output	Video Monitoring
storage	Storage Settings
record	Recording Settings
performance	Performance Indicators
ntp	NTP Time Synchronization
webrtc	WebRTC
playback	Playback Module
system	Basic Configuration
network	Network Settings
firmware	Firmware Module

1. Configuration

1.1 Get General Configuration Information

Description: Get general configuration information of the software

API URL

```
/api/r1/config/general/get.json
```

Request

```
Method: GET
Parameters: NONE
```

Example:

```
GET /api/r1/config/general/get.json HTTP/1.1
Accept: application/json, text/plain, */*
```

Response

Example:

```
{
   "result": "ok",
   "data": {
      "deviceName": {
         "zh": "Cube R1",
         "en": "Cube R1"
      },
      "languageShow": false,
      "websocketPort": 12346,
      "languageOptions": [
         {
            "value": "zh",
            "name": "简体中文"
         },
         {
            "value": "en",
            "name": "English"
         }
      ]
   }
}
```

Field	Value	Description
deviceName	[STRING]	Current software name, inherited from the encoding and decoding device due to historical reasons
languageShow	[BOOLEAN]	Whether language switch is displayed on the interface
languageOptions	[ARRAY]	Supported language list for interface display
languageOptions[0].value	[STRING]	Value of language
languageOptions[0].name	[STRING]	Text displayed for language
languageOptions[0].default	[BOOLEAN]	Default language for interface
websocketPort	[INTEGER]	WebSocket port

2. User Management Module

Description: User login, logout, and password modification

2.1 Login

API URL

```
api/r1/users/login.json
```

Request

```
Method：POST
```

Parameter	Value	Description
username	[STR] ,Required	User name
password	[STR] ,Required	User password

```
{
"username":"admin",
      "password":"admin"
}
```

Response

Example：

```
{
        "data": {
                "alias": "admin",
                "changed": true,
                "token": "3ee69a2f13b71cc57061845bc3bf04c5",
                "username": "admin"
        },
        "result": "ok"
}
{
        "msg": "Username or password is incorrect",
        "result": "error"
}
{
        "result": "error"
}
```

2.2 Logout

API URL

```
api/r1/users/logout.json
```

Request

Method：GET

Response

Example：

```
{
        "result": "ok"
}
{
        "msg": "Token.Error",
        "result": "error"
}
```

2.3 Change Password

API URL

```
api/r1/users/modify.json
```

Request

Method：POST

Parameter	Value	Description
username	[STR] ,Required	Username
password	[STR] ,Required	New password
oldpassword	[STR] ,Required	Original password

```
{
"oldpassword":"admin",
        "username":"admin",
        "password":"Admin1"
}
```

Response

Example：

```
{
    "result": "ok"
}
{
    "msg": "Token.Error",
    "result": "error"
}
```

3. Source Discovery

Description: Mainly includes adding, deleting, and modifying operations for the automatic and manual discovery of source lists.

3.1 Get Source Discovery List

API URL

```
/api/r1/source/list.json
```

Response

```
Method： GET
```

Example：

```
{
    "data": [
        {
            "disc_id": "1",
            "disc_name": "auto-discovery",
            "sources": [
                {
                    "id": "DECODER-00000000 (1_channel)",
                    "ip": "192.168.39.191:5961",
                    "name": "DECODER-00000000 (1_channel)"
                },
                {
                    "id": "HOLLY (VLC 1)",
                    "ip": "192.168.39.168:5962",
                    "name": "HOLLY (VLC 1)"
                },
                {
                    "id": "N6-00000000 (Channel-HX)",
                    "ip": "192.168.39.17:5964",
                    "name": "N6-00000000 (Channel-HX)"
                }
            ]
        },
        {
            "disc_id": "5",
            "disc_name": "43",
            "sources": [
                {
                    "id": "N20-24 (Channel-1)",
                    "ip": "192.168.43.24:5961",
                    "name": "N20-24 (Channel-1)"
                }
            ]
        }
    ],
    "result": "ok"
}
```

Field	Value	Description
disc_id	[STRING]	ID of source discovery
disc_name	[STRING]	Name of source discovery
id	[STRING]	ID of a source item in the source discovery
name	[STRING]	Name of a source item in the source discovery
ip	[STRING]	IP: Port of a source item in the source discovery
official	[BOOL]	Whether it is an official device, not supported for recognition at present

3.2 Add a group

Description: Add a group, the name cannot be the same as other groups.

API URL

```
/api/r1/source/add.json
```

Request

```
Method：POST
```

Example：

Automatic discovery

```
{
"disc_name": "自动发现 1",
  "disc_type": "Auto",
  "discovery_enable": true,
  "discovery_ip": "192.168.1.1",
  "group": ["kiloview", "NDI","test"]
}
```

Manual discovery

```
{
"disc_name": "手动添加 1",
  "disc_type": "Manual",
  "group": ["kiloview", "NDI","test"],
  "ip": ["192.168.40.11", "192.168.40.12"]
}
```

Data Field Explanation:

Field	Value	Description
disc_name	[STRING]	Source discovery name
disc_type	[STRING]	Source discovery type: divided into automatic and manual types:
discovery_enable	[BOOLEAN]	Automatic (Auto)
discovery_ip	[STRING]	Manual (Manual)

group	[ARRAY]	Whether to enable discovery server
ip	[ARRAY]	Discovery server address

Response

Example：

```
{
   "result":"ok"
}
```

3.3 Get Group Parameters

Description： Modify the name, type, discovery server, discovery group list or IP list of the current source discovery.

API URL

```
/api/r1/source/get_param.json
```

Request

```
Method：  GET
```

Response

Format（Example）：

Automatic discovery

```
{
"data":{
"disc_id": "1",
"disc_name": "自动添加 1",
   "disc_type": "Manual" ,
   "group": ["kiloview", "NDI","test"],
   "ip": ["192.168.40.11", "192.168.40.12"]
   },
"result": "ok"
}
```

Manual discovery

```
{
"data":{
"disc_id": "1",
"disc_name": "手动添加 1",
   "disc_type": "Auto" ,
   "discovery_enable": true,
   "discovery_ip": "192.168.1.1",
   "group": ["kiloview", "NDI","test"]
   },
"result": "ok"
}
```

3.4 Modify Group Parameters

Description： Modify the name, type, discovery server, discovery group list or IP list of the current source discovery.

API URL

```
/api/r1/source/modify.json
```

Request

Method： POST

Example：

Automatic discovery

```
{
"disc_id": "1",
"disc_name": "自动发现 1",
   "disc_type": "Auto",
   "discovery_enable": true,
   "discovery_ip": "192.168.1.1",
   "group": ["kiloview", "NDI","test"]
}
```

Manual discovery

```
{
"disc_id": "1",
"disc_name": "手动发现 1",
   "disc_type": "Manual",
   "group": ["kiloview", "NDI","test"]，
   "ip": ["192.168.40.11", "192.168.40.12"]
}
```

Response

Format（Example）：

```
{
   "result":"ok"
}
```

3.5 Delete a group

API URL

/api/r1/source/remove.json

Request

Method： **POST**

Format（Example）：

```
{
      "disc_id": "1"
}
```

Field	**Value**	**Description**
disc_id	**[STRING]**	ID of source discovery

Response

Example：

```
{
   "result":"ok"
}
```

3.6 Refresh Discovery List

API URL

/api/r1/source/refresh.json

Request

Method： POST

Response

Example:

```
{
  "refresh":true
}
```

Field	**Value**	**Description**
refresh	**[BOOLEAN]**	Whether to refresh the discovery list once

Response

Example:

```
{
"result":"ok"
}
```

4. Layout Settings

4.1 Get Layout Icon Parameter List

API URL

```
/api/r1/layout/icon.json
```

Request

```
Method: GET
Parameters: NONE
```

Example:

```
GET /api/output/icon.json HTTP/1.1
Accept: application/json, text/plain, */*
```

Response

Example:

```
{
  "result": "ok",
  "data": [
    {
      "id": 1,
      "number": "1",
      "position": [
        {
          "x": 0,
          "y": 0,
          "width": 100,
          "height": 100
        }
      ]
    }
  ]
}
```

4.2 Set current layout

API URL

```
/api/r1/layout/setLayout.json
```

Request

```
Method: POST
```

Parameter	**Value**	**Description**
project_id	**[INT], Required**	Project ID
layout_id	**[INT], Required**	Layout ID
layout_number	**[INT], Required**	Split-screen number of layout

Example:

```
POST /api/r1/layout/setLayout.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
   "project_id": 1,
   "layout_id": 3,
   "layout_number": 9
}
```

Response

Example:

```
{
"result":"ok"
}
```

5. output Video Monitoring

5.1 Get Current Monitoring Video List

API URL

```
/api/r1/output/get.json
```

Request

```
Method: GET
```

Example:

```
GET /api/r1/output/get.json?project_id=1 HTTP/1.1
Accept: application/json, text/plain, */*
```

Response

Example:

```
{
   "result": "ok",
   "data": {
       "layout_id": "1",
       "layout_number": 1,
       "layout": [
{
             "id": 1,
             "name": "test",
             "stream_id": "KILOVIEW (Chan 1 )",
             "stream_name": "KILOVIEW (Chan 1 )",
             "bit_rate":40960,
             "ntp_state":"sync",
"showAll": true,
             "showVolume": true,
             "showVideo": true,
"isKiloview":true,
"volume": "off"
          }
       ]
   }
}
```

Field	Value	Description
id	**[INT]**	The position of the video in the panel
bit_rate	**[INT]**	Video bit rate in bits
ntp_state	**[STRING]**	NTP status unsync: not synchronized, error: synchronizing, sync: synchronization exception
isKiloview	**[BOOLEAN]**	Whether it is an official device true or false
showAll	**[BOOLEAN]**	Whether to display speakers and screen
showVolume	**[BOOLEAN]**	Whether to display speakers
showVideo	**[BOOLEAN]**	Whether to display screen

Official device: equipment or product produced by Kiloview or NewTek

5.2 Set Monitoring Video Source

Description：Set a video source to a split-screen window

API URL

```
/api/r1/output/setSource.json
```

Request

```
Method: POST
```

Parameter	**Value**	**Description**
project_id	**[STRING], Required**	Project ID
position	**[INT], Required**	Output position in layout 1-20
stream_id	**[INT], Required**	NDI Stream ID 1-20
type	**[STRING], Required**	Operation type source: Set a video source to the window display: Swap the video sources between two windows

Example:

```
POST /api/output/setSource.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
   "project_id": 1,
   "position": 3,
   "stream_id": "KILOVIEW wersdfdsdf",
   "type": "source"
}
```

Response

Normal situation

```
{
"result":"ok"
}
```

Exceeding permissions

```
{
"result":"error",
"reason":"out-of-auth"
}
```

5.3 Modify Monitoring Window Name

Description： Modify the name of a split-screen window

API URL

```
/api/r1/output/setName.json
```

Request

```
Method: POST
```

Parameter	**Value**	**Description**
project_id	**[STRING], Required**	Project ID

position	**[INT], Required**	Output position in layout 1-20
stream_id	**[INT], Required**	NDI Stream ID 1-20

Example:

```
POST /api/r1/output/setSource.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
   "project_id": 1,
   "position": 3,
   "name": "修改后的名称"
}
```

5.4 Turn on and off video and audio of a monitoring window

Description：Control the video and speakers of a split-screen window

API URL

```
/api/r1/output/setEnable.json
```

Request

```
Method: POST
```

Parameter	**Value**	**Description**
project_id	**[INT], Required**	Current project ID
position	**[INT], Required**	Select the location to control monitoring
showAll	**[BOOLEAN], Required**	Display speakers and video (select all button)
showVolume	**[BOOLEAN], Required**	Display speakers
showVideo	**[BOOLEAN], Required**	Display screen

Example:

```
POST /api/r1/output/setEnable.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
   "project_id": 1,
   "position": 3,
   "showAll":true,
   "showVolume":true,
   "showVideo":true
}
```

Response

Example:

```
{
"result":"ok"
}
```

5.5 Remove a Monitoring Video

Description：Remove a video source from a split-screen window.

API URL

```
/api/r1/output/remove.json
```

Request

```
Method: POST
```

Parameter	**Value**	**Description**
project_id	**[INT], Required**	Current project ID
position	**[INT], Required**	Select the location to control monitoring

Example:

```
POST /api/output/remove.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
   "project_id": 1,
   "position": 1
}
```

Response

Example:

```
{
"result":"ok"
}
```

5.6 Clear All Monitoring Videos

Description：Clear all video sources in the split-screen windows of the current layout.

API URL

```
/api/r1/output/deleteAll.json
```

Request

```
Method: POST
```

Parameter	**Value**	**Description**
project_id	**[INT], Required**	Current project ID

Example:

```
POST /api/output/deleteAll.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
"project_id": 1
}
```

Response

Example:

```
{
"result":"ok"
}
```

5.7 Set whether to display all screens

Description： Control whether the video in all split-screen windows of the current layout is displayed.

API URL

```
/api/r1/output/setGlobalVisible.json
```

Request

```
Method: POST
```

Parameter	**Value**	**Description**
project_id	**[INT], Required**	Current project ID
outputGlobalVisible	**[BOOLEAN], Required**	Whether to display all screens

Example:

```
POST /api/r1/output/setGlobalVisible.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
   "project_id": 1,
   "outputGlobalVisible": false
}
```

Response

Example:

```
{
"result":"ok"
}
```

5.8 Set whether the current window is muted

Description: Control whether the video of the current window is muted

API URL

```
/api/r1/output/window_audio_set.json
```

Request

```
Method: POST
```

Parameter	**Value**	**Description**
project_id	**[INT]**	Current project ID, default is 1
position	**[INT], Required**	Current position ID
volume	**[BOOLEAN], Required**	Sound switch: on off

Example:

```
POST /api/output/setGlobalVisible.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
"project_id":1,
      "position":6,
      "volume":"on"
}
```

Response

Example:

```
{
"result":"ok"
}
```

6. Storage Settings

6.1 Obtain Storage Information

Description: Obtain basic disk data, split type, and selected start disk.

API URL

```
/api/r1/storage/getStoInfo.json
```

Request

```
Method: GET
Parameters: NONE
```

Example:

```
GET /api/r1/storage/get.json HTTP/1.1
Accept: application/json, text/plain, */*
```

Response

Exampl

```
{
    "result": "ok",
    "data": {
        "disk": [{
                "name": "disk1",
                "id": "0",
                "unlock": true,
                "state": "online",
                "total": "900G",
                "used": "30M",
            "rate": 1,
                "speed": 0,
                "recoding": true,
                "msg": "",
                "channel": [{
                        "id": "111333",
                        "name": "Camera1"
                    },
                    {
                        "id": "1113323",
                        "name": "Camera2"
                    }
                ]
            },
            {
                "name": "disk2",
        "id": "1",
        "unlock": false,
        "state": "offline",
        "total": "",
        "used": "",
        "rate": 1,
        "speed": 0,
        "recoding": false,
        "msg": "",
        "channel": [{
            "id": "111333",
            "name": "Camera1"
          },
          {
            "id": "1113323",
            "name": "Camera2"
          }
        ]
      }
    ],
    "choose": "1",
    "limitType": "size",
    "limitSize": 10,
    "limitTime": 60
  }
}
```

Field	Value	Description
id	**[INT]**	Disk unique value
total	**[STR]**	Total disk capacity
used	**[STR]**	Disk used capacity
rate	**[INT]**	Disk storage percentage

6.2 Set storage parameters

API URL

/api/r1/storage/setStoLimit.json

Request

Method: POST

Parameter	**Value**	**Description**
choose	**[STRING], Required**	Start disk setting
limitType	**[STRING], Required**	Recording file split rules, support size and time rules, size, time, default
limitSize	**[INT]**	Size limit, unit GB

limitTime	**[INT]**	Time limit, unit Min

Example:

```
POST /api/r1/storage/set.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
"choose": 1,
"limitType":"size",
"limitSize": 10,
"limitTime": 60
}
```

Response

Example:

```
{
"result":"ok"
}
```

6.3 When the lock state is updated

Returned via WebSocket

```
{
	"socketType":"storePathUpdate",
	"result":"ok",
	"data":{
		"disk1": "unlock",
		"disk2": "lock"
	}
}
```

7. Recording settings

The button controls the recording of all video windows in the project, but each recording is a separate thread, and its return result is an asynchronous event, so it is notified to the page through the WebSocket.

7.1 Get recording status

API URL

```
/api/r1/record/getRecStatus.json
```

Request

```
Method: GET
```

Parameter	**Value**	**Description**
project_id	**[INT], Required**	Project id

Example:

```
GET /api/r1/record/get.json?project_id=1 HTTP/1.1
```

```
Accept: application/json, text/plain, */*
```

Response

Example:

```
{
   "result":"ok",
   "data":{
      "isRecording":true,
   "startTime":1622100648254 // Starting recording timestamp
      "msg":""
   }
}
```

7.2 Set recording status

API URL

```
/api/r1/record/setRecStatus.json
```

Request

```
Method: POST
```

Parameter	Value	Description
isRecording	**[BOOLEAN], Required**	Whether recording is in progress

Note: it must be in a ready state to start recording. After stopping recording, it can exit the ready state.

Example:

```
POST /api/r1/record/set.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
"isRecording": true
}
```

Response

Example:

```
{
"result":"ok"
}
```

7.2.1 Recording update

websocket

```
{
"update": "record",
"result": "ok"
}
```

7.3 Get recording settings information

API URL

```
/api/r1/record/getinfo.json
```

Request

```
Method: GET
```

Example:

```
GET /api/r1/record/get.json?project_id=1 HTTP/1.1
Accept: application/json, text/plain, */*
```

Response

Example:

```
{
   "result":"ok",
   "data":{
      "solution": 0, //0:Native     1: h.264     2: h.265
      "sync": ture,
      "isStart":true,
      "startTime":"2023-04-07 14:34:33",
      "isStop":false,
   "stopTime":"2023-04-07 16:34:33"
   }
}
```

Parameter	**Value**	**Description**
solution	**[INT], Required**	Encoding method: 0 1 2 0 native 1 H.264 2 H.265
sync	**[BOOL], Required**	Whether to force time synchronization
isStart	**[BOOL], Required**	Whether to schedule recording
startTime	**[STR], Required**	Scheduled recording start time
isStop	**[BOOL], Required**	Whether to schedule recording stop
stopTime	**[STR], Required**	Scheduled recording stop time

7.4 Set recording settings information

API URL

```
/api/r1/record/setinfo.json
```

Request

```
Method: POST
```

Example:

```
{
      "solution": 0, //0 native    1 H.264    2 H.265
```

```
        "sync": ture,
"isStart":true,
        "startTime":"2023-04-07 14:34:33",
        "isStop":false,
    "stopTime":"2023-04-07 16:34:33"
  }
```

Response

Example:

```
{
    "result":"ok",
}
```

8. Performance indicators

8.1 Obtain system performance parameters

API URL

```
/api/r1/performance/getSys.json
```

Request

```
Method: GET
Parameters: NONE
```

Example:

```
GET /api/r1/performance/getSys.json HTTP/1.1
Accept: application/json, text/plain, */*
```

Response

Example:

```
{
"result":"ok",
"data":{
"cpu": 45,
"gpu": 45,
"temp": 90,
"mem_use": 2560,
"mem_total": 8192,
"net_up": 1024,
"net_down": 128
}
}
```

Field	**Value**	**Description**
cpu	**[DOUBLE]**	CPU usage percentage, 0-100
gpu	**[DOUBLE]**	GPU usage percentage, 0-100
temp	**[INT]**	CPU temperature, Celsius
mem_use	**[INT]**	Memory usage, unit MB
mem_total	**[INT]**	Total memory, unit MB
net_up	**[INT]**	Upstream network speed, unit KB
net_down	**[INT]**	Downstream network speed, unit

		KB

9. webrtc

9.1 get_sdp

API URL

```
/api/r1/webrtc/get_sdp.json
```

Request

```
/api/r1/webrtc/get_sdp.json?pro_id=1&win_id=1&ip=192.168.39.191
```

Parameter	Value	Description
ip	[STRING], Required	IP address
pro_id		Project ID, reserved, always 1
win_id		Window ID, reserved, always 1

Response

Example:

```
{
	"data": {
		"sdp": "v=0\r\no=- 1495799811084970 2 IN IP4 192.168.39.191\r\ns=-\r\nt=0 0\r\na=group:BUNDLE video\r\na=msid-semantic: WMS janus\r\na=ice-lite \r\nm=video 9 UDP/TLS/RTP/SAVPF 126\r\nc=IN IP4 0.0.0.0\r\na=ice-ufrag:0n1s\r\na=ice-pwd:btqo6kj056t9xomlgu3xzd89\r\na=ice-options:trickle\r\na=fingerprint:sha-256 8F:08:27:BA:7A:8A:02:98:8E:5B:0C:BD:62:9A:F3:14:5C:FF:E5:5B:7B:75:E3:3A:85:72:54:2E:7B:EF:AD:16\r\na=setup:actpass\r\na=mid:video\r\na=sendonly\r\na=rtcp-mux\r\na=rtpmap:126 H264/90000\r\na=fmtp:126 level-asymmetry-allowed=1;packetization-mode=1;profile-level-id=42e01f\r\na=candidate:1 1 udp 12345678 192.168.39.191 51498 typ host\r\na=end-of-candidate\r\n",
		"session_id": "1"
	},
	"result": "ok"
}
```

9.2 remove_session

API URL

```
/api/r1/webrtc/remove_session.json
```

Request

```
/api/r1/webrtc/remove_session.json?session_id=1
```

Parameter	Value	Description
session_id	[STRING], Required	Session id

Response

Example:

```
{
	"result": "ok"
}
```

10. Playback module

10.1 Get playback video list

API URL

```
/api/r1/playback/getPlayList.json
```

Request

```
Method: POST
```

Example:

```
{
   "disk_id": "1",
   "channelName": "source-1",
   "startTime": "2023-4-10 13:09:39",
   "stopTime": ""
}
```

Response

Example:

```
{
   "result": "ok",
   "data": [{
                        "name": "14022023-09.48.21",
                        "videoSrc": "/data/Camera1/big_buck_bunny.mp4",
                    }
                ]
}
```

Field	Value	Description
disk_id	[STRING]	Disk ID
channelName	[STRING]	Channel name
startTime	[STRING]	Filter start time, blank means no start time limit
stopTime	[STRING]	Filter end time, blank means no end time limit

10.2 Delete single video

API URL

```
/api/r1/playback/remove.json
```

Request

```
Method: POST
```

Example:

```
{
   "videoSrc": "/media/ndicore/309E-8297/source-1/1.mov"
}
```

Response

Example:

```
{
   "result": "ok"
}
```

10.3 Get single video information

API URL

```
/api/r1/playback/getVideoInfo.json
```

Request

```
Method: POST
```

Example:

```
{
   "videoSrc":"/media/ndicore/309E-8297/Camera-2/16052023-06.31.53-1.mov"
}
```

Response

Example:

```
{
   "result": "ok",
   "data": {
                    "name": "14022023-09.48.21",
                    "filetype": "mov",
                    "size": 4067211,
                    "createTime": 1601412123,
                    "duration": "00:09:15",
                    "frameWidth": "1920",
                    "frameHeight": "1080",
                    "dataRate": 124511,
                    "totalBitrate": 2412225,
                    "frameRate": 30,
                    "audioBitrate": 20000,
                    "audioChannel": 2,
                    "audioSample": 48000
                 }
}
```

11. System settings

Field	Value	Description
time	[STRING]	Current device time
timetype	[STRING]	Time synchronization method: "dev" "ntp" Synchronize with device: dev Synchronize with ntp server: ntp
ntp	[STRING]	NTP server address
Timezone	[STRING]	Current selected time zone

hostname	[STRING]	Device name
language	[STRING]	Current selected language: zh en

11.1 Get host name

API URL

```
/api/r1/system/getHostname.json
```

Response

Example:

```
{
        "result": "ok",
        "data": {
                "hostname": "CubeR1"
        }
}
```

11.2 Modify host name

API URL

```
/api/r1/system/setHostname.json
```

Request

```
{
"hostname":"cube"
}
```

Response

Example:

```
{
    "result": "ok"
}
```

11.3 Get current time zone

API URL

```
/api/r1/system/getTimeZone.json
```

Response

Example:

```
{
        "result": "ok",
        "data": {
      "timezone": "Asia/Aqtobe",
"offset": "5"
        }
}
```

11.4 Modify current time zone

API URL

```
/api/r1/system/setTimeZone.json
```

Request

```
{
"timezone": "Asia/Aqtobe",
"offset": "5"
}
```

Response

Example:

```
{
   "result": "ok"
}
```

11.5 Get time synchronization method

API URL

```
/api/r1/system/getTimer.json
```

Response

Example:

```
{
      "result": "ok",
      "data": {
            "time": "1698989621283 ",
            "timeType": "dev",
            "ip": "192.168.1.168",
      }
}
```

11.6 Modify time synchronization method

API URL

```
/api/r1/system/setTimer.json
```

Request

```
{
"timeType": "ntp",
"ip": "192.168.1.168"
}
```

Response

Example:

```
{
   "result": "ok"
}
```

11.7 One-click time synchronization

API URL

```
/api/r1/system/timer/apply.json
```

Request

```
Method: POST
```

Example:

```
POST /api/r1/ntp/apply.json HTTP/1.1
Accept: application/json, text/plain, */*
Content-Type: application/json

{
project_id:1,
}
```

Response

Example:

```
{
"result":"ok"
}
```

12. Network settings

12.1 Obtain network information

API URL

```
/api/r1/network/getNet.json
```

Response

Example:

```
{
"result":"ok",
"data":[
        {
        "dynamic":"y",
        "dns":"223.5.5.5",
        "device":"eth0",
        "ip":"192.168.39.96",
        "state":"up","mac": "32:30:11:11:11:ad",
        "gw":"192.168.39.254",
        "sub":"255.255.255.0"},

        {
        "dynamic":"y",
        "dns":"223.5.5.5",
        "device":"ofp0", //光纤口 1 标识，具体待定
        "ip":"0.0.0.0",
        "state":"down","mac": "32:30:11:11:11:ae",
        "gw":"192.168.39.254",
        "sub":"255.255.255.0"},
        {
        "dynamic":"n",
        "dns":"223.5.5.5",
        "device":"ofp1",
        "ip":"192.168.39.98",
        "state":"up","mac": "32:30:11:11:11:af",
        "gw":"192.168.39.254",
        "sub":"255.255.255.0"}
        ]
}
```

12.2 Modify network configuration

API URL

```
/api/r1/network/modifyNet.json
```

Request

```
        {
        "device":"ofp1",
        "dynamic":"n",
        "dns":"223.5.5.5",
        "ip":"192.168.39.198",
        "state":"up",
        "gw":"192.168.39.254",
        "sub":"255.255.255.0"
        }
```

Response

Example:

```
{
    "result": "ok"
}
```

13. Firmware settings

13.1 Obtain version information

API URL

```
/api/r1/firmware/get.json
```

Response

Example:

```
{
"result":"ok",
"data":
{
        "softwareVersion":"1.0.0",
	"firmwareVersion":"1.0.0"
}
}
```