Leading Solution Provider for IP-based Video Transmission

API

N60 Bi-Directional NDI Converter

NDI High Bandwidth+NDI|HX

(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® 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

1 System Module 5
1.1 User Management 5
1.1.1 User Login 5
1.1.2 Login Verification 6
1.1.3 Get All Users 6
1.1.4 Add User 7
1.1.5 Delete User (can be single or batch deletion) 8
1.1.6 Modify User Information 8
1.2 System Time Zone 9
1.2.1 Set Location Time Zone 9
1.2.2 Set System Time 9
1.3 System Configuration 10
1.3.1 Get Open Source License 10
1.3.2 Get Device Name 11
1.3.3 Set Device Name 11
1.3.4 Get HTTP and HTTPS port switch 12
1.3.5 Set HTTP and HTTPS port switch 12
1.3.6 Factory Reset 13
1.3.7 Reset 13
1.3.8 Reboot Device 14
1.3.9 Get Network Card Speed 14
1.3.10 Get System Information 15
1.3.11 Set device serial number 16
1.3.12 Upload firmware upgrade 16
1.3.13 Switch for testing page service 17
1.3.14 Status of the testing page service 17
2 Video encoding and decoding 18
2.1 Decoding 18
2.1.1 Selecting the current decoding stream 18
2.1.2 Selecting a decoding stream from the preset 19
2.1.3 Get decoding information 19
2.1.4 Setting the output decoding information 23
2.1.5 Setting NDI connection mode 25
2.2 Discovery Service Module 26
2.2.1 Set Discovery Server 26
2.2.2 Get discovery server 26
2.2.3 Scan and retrieve NDI Source list 26
2.2.4 Manually add source configuration 28
2.2.5 Get manually added NDI source addresses 28
2.2.6 Setting discovery parameters in decoding mode 29
2.2.7 Getting the discovery parameters for the device in decoding mode 29
2.3 Preset module 30
2.3.1 Get the preset list 30
2.3.2 Add preset 32
2.3.3 Delete preset 32
2.3.4 Set background color 33
2.4 Decoding PTZ camera control 33
2.4.1 Decoding mode PTZ control 33
2.5 Encoding 34
2.5.1 Get the names of all encoding objects 34
2.5.2 Encoding stream switch 35
2.5.3 Get encoding stream encoding parameters 35
2.5.4 Get video source parameters 37

2.5.5 Set main stream (NDI|HX) encoding parameters 38
2.5.6 Set FULL NDI stream (NDI|HB) encoding parameters 39
2.5.7 Get stream information 39
2.5.8 Re-stream all streams 40
2.5.9 Get stream-related properties and configurations 41
2.5.10 Get real-time bitrate and connection count of the stream 42
2.5.11 Update stream-related configuration 43
2.6 Mode 44
2.6.1 Set Working Mode (Encoding Mode | Decoding Mode) 44
2.6.2 Get Working Mode 45
2.7 Encode PTZ Control 45
2.7.1 Encode PTZ Control 45
2.7.2 Get PTZ Information 46
2.7.3 Get PTZ Protocol 47
2.7.4 Get Serial Devices 48
2.7.5 Modify PTZ Information 48
2.8 QR Code Scanning 49
2.8.1 Virtual User Identity Login 49
2.8.2 Get QR Code ID 50
2.8.3 QR Code Verification 50
2.8.4 Check if the Current Mode is Decoding 51
2.8.5 Get QR Code 52
3 Network 52
3.1 Network Management 52
3.1.1 Get All Network Information 52
3.1.2 Modify current device network information 54
3.1.3 Ethernet switch 55
3.1.4 Get network card information 55
3.1.5 Set network card information 56
3.1.6 Set link aggregation 57
3.1.7 Get link aggregation information 58
3.1.8 Restart MDNS 59
3.2 Static route management 59
3.2.1 Add Static Route 59
3.2.2 Get static route list 60
3.2.3 Modify static route 60
3.2.4 Delete static route 61
3.2.5 Static route switch 62
4 KiloLink 62
4.1 KiloLink Client Service 62
4.1.1 Get Current Device Network Interfaces 62
4.1.2 Get KiloLink client settings 63
4.1.3 Set KiloLink client configuration parameters 63
4.1.4 Validate token 64
5、Voice Intercom 64
5.1 Talk server 64
5.1.1 Add Device 64
5.1.2 Set connection between local device and target device 65
5.1.3 Set connection between web page and target device 66
5.1.4 Set connection between device and voice server 66
5.1.5 Query device connection status 67
5.1.6 Query device list 67
5.1.7 Delete device 68
5.1.8 Query serial number of local device 69

5.1.9 Talk server version 69
5.1.10 Set language..... 70
5.1.11 Notify device for call deletion (interface not accessible to the front-end)..... 70
5.1.12 Device request connection, device-to-device connection (interface not accessible to the front-end)
..... 71

1 System Module

1.1 User Management

1.1.1 User Login

- Interface Description

Requesting this interface alone does not directly log in. You need to take the token and alias values returned by the interface, combine them with the username to construct a cookie, and include it in the request header to request the validation interface 1.1.2. Only after successful validation can you log in.

- Request URL

  /api/systemctrl/users/login

- Request Method

  POST

- Request Parameters（Example）

```
{
    "username": "admin",
    "password": "Admin123"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
App	{"language":"en"}	Y	**[STRING]**	Request header parameter. The current version only supports Chinese and English languages, which are "en" / "zh" respectively, used for multilingual output of popup prompts. **Note:** In the current version, if you want to get the correct error message when requesting this interface alone, please be sure to include this parameter in the request header. Otherwise, the msg field in the returned prompt will be null.
username	"admin"	Y	**[STRING]**	Username
password	"Admin123"	Y	**[STRING]**	Password

- Response Parameters（Example）

```
{
    "result": "ok",
    "data": {
        "token": "0621da14f9d6546688cf87c5eab9c2f9",
        "alias": "Admin",
        "changed": true
    },
    "msg": null
}
```

- Response Parameter Description (Explanation of the data field)

Field Name	Type	Field Description
token	**[STRING]**	Token issued during login, used for login

		persistence
alias	**[STRING]**	Alias
changed	**[BOOLEAN]**	Whether the user has changed the initial password. If not changed, the front-end will redirect to the page for changing the initial password.

1.1.2 Login Verification

- Interface Description

Used for normal login requests and session verification and validation for both normal login requests and requests from the KiloLink reverse proxy. For normal login requests, the cookie needs to be added in the request header along with other field values, as shown in the example. For KiloLink reverse proxy requests, the "platform" field needs to be added in the request and the corresponding field identifier should be specified, as shown in the example.

- Request URL
  /api/systemctrl/users/session
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
platform	"this-is-made-by-kiloview-for-platform-login"	Y	**[STRING]**	Request Parameters: If users want to use this interface separately, they need to add the value individually to the request header of the KiloLink server.
Cookie	"language=zh; user=kiloview; alias=Kiloview; token=18532c790a58dcc31b0d473b3a791770"	Y	**[STRING]**	Request header parameter: If you want to use this interface separately, the user needs to add this value to the request header individually.

- Response Parameters (Example）

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

- Description on Response Parameter
  None

1.1.3 Get All Users

- Interface Description

This interface is used to retrieve all the users that can log in to the current device.

- Request URL
  /api/systemctrl/users/list
- Request Method
  GET
- Request Parameters (Example）

None

- Request Parameter Description

None

- Response Parameters (Example）

```
{
    "result": "ok",
    "data": [
        {
            "id": "admin",
            "alias": "Admin",
            "web": true,
            "api": true,
            "create_time": "1970-01-01 00:00:00"
        }
    ]
}
```

- Response Parameter Description (Explanation of individual fields in the "data" field)

Field Name	Type	Field Description
id	**[STRING]**	Username
alias	**[STRING]**	Alias
web	**[BOOLEAN]**	Whether web-based user creation is allowed
api	**[BOOLEAN]**	-
create_time	**[STRING]**	Creation time

1.1.4 Add User

- Interface Description

This interface is used by administrators or other logged-in users to add new users.

- Request URL

/api/systemctrl/users/add

- Request Method

POST

- Request Parameters (Example)

```
{
    "id": "test_user",
    "password": "qwer1234",
    "confirmNewPassword": "qwer1234",
    "alias": "test_user",
    "enable_web": true,
    "enable_api":true
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
id	"test_user"	Y	**[STRING]**	Username
alias	"test_user"	Y	**[STRING]**	Alias
password	"qwer1234"	Y	**[STRING]**	User password
confirmNewPassword	"qwer1234"	Y	**[STRING]**	Confirm password
enable_web	true	Y	**[BOOLEAN]**	Whether web-based user creation is allowed
enable_api	true	Y	**[BOOLEAN]**	-

- Response Parameters (Example)

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

- Response Parameter Description

None

1.1.5 Delete User (can be single or batch deletion)

- Interface Description

When requesting this interface, pass in a single username or a list of usernames, and then the corresponding user can be deleted.

- Request URL

/api/systemctrl/users/remove

- Request Method

POST

- Request Parameters(Example)

Single deletion

```
{
   "ids": "Jack"
}
```

Batch deletion

```
{
   "ids": ["Jack", "Bob"]
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
ids	"Jack" / ["Jack", "Bob"]	Y	**[STRING]/[ARRAY(STRING)]**	Username or List of usernames

- Response Parameters (Example)

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

- Response Parameter Description

None

1.1.6 Modify User Information

- Interface Description

By requesting this interface, you can modify user information other than what is already saved on the device, but only the nickname can be modified.

- Request URL

/api/systemctrl/users/modify

- Request Method

POST

- Request Parameters (Example)

```
{
      "id": "test1",
      "alias": "test11",
      "enable_web": true,
      "enable_api":true
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
id	"test1"	Y	**[STRING]**	Username
alias	"test11"	Y	**[STRING]**	Alias
enable_web	true	Y	**[BOOLEAN]**	Whether web-based user creation is allowed
enable_api	true	Y	**[BOOLEAN]**	-

- Response Parameters (Example)

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

- Response Parameter Description

None

1.2 System Time Zone

1.2.1 Set Location Time Zone

- Interface Description

Request this interface to set the location time zone by passing in the standard time string of the specified location.

- Request URL

/api/systemctrl/systime/setLocation

- Request Method

POST

- Request Parameters (Example)

```
{
    "timezone": "Asia/Shanghai",
    "offset": 8,
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
timezone	"Asia/Shanghai"	Y	**[STRING]**	Time zone
offset	8	N	**[INT]**	If the timezone cannot be found, this field will be used. This field represents the hours added or subtracted from Greenwich Mean Time, such as GMT+8, ranging from -14 to 12.

- Response Parameters (Example)

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

- Response Parameter Description

None

1.2.2 Set System Time

- Interface Description

Request this interface to set the system time. You can choose to synchronize with the current PC, manually adjust the time, or synchronize from an NTP server.

- Request URL

/api/systemctrl/systime/setTime

- Request Method

POST

- Request Parameters (Example)
  None
- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
timetype	"pc"	Y	**[STRING]**	Time setting method: pc: synchronize with the current PC; manual: adjust the time manually; ntp: synchronize from NTP server
time	"2023-08-07 17:37:14"	N	**[STRING]**	The standard format of time is "Y-m-d H:M:S". Required when the timetype parameter value is pc or manual!
ntp	"ntp1.aliyun.com ntp2.aliyun.com ntp3.aliyun.com"	N	**[STRING]**	NTP server address, multiple servers are separated by spaces. Required when the timetype parameter value is ntp!

- Response Parameters (Example)

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

- Response Parameter Description
  None

1.3 System Configuration

1.3.1 Get Open Source License

- Interface Description

Request this interface to obtain the open source license, a fixed value.

- Request URL
  /api/systemctrl/system/components
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameter

```
[{
    "name": "vue",
    "url": "MIT",
    "software": "MIT License",
    "address": "https://github.com/vuejs/vue"
}, {
    "name": "Element-UI",
    "url": "MIT",
    "software": "MIT License",
    "address": "https://github.com/ElemeFE/element"
},
    {
        "name": "python3",
        "url": "PSF",
        "software": "PSF",
```

```
        "address": "http://www.python.org"
    }, {
        "name": "nginx",
        "url": "BSD2C",
        "software": "BSD-2c",
        "address": "http://nginx.org"
    }, {
        "name": "libev",
        "url": "BSD2C",
        "url2": "GPL",
        "software": "BSD-2c",
        "software2": "GPLv2+",
        "address": "http://dist.schmorp.de/libev"

    }, {
        "name": "openssl",
        "url": "OpenSSL",
        "url2": "SSLeay",
        "software": "OpenSSL",
        "software2": "SSLeay",
        "address": "http://www.openssl.org"
    }]
```


- Response Parameter Description
  None

1.3.2 Get Device Name

- Interface Description

Request this interface to obtain the name of the device, which is usually specified by the user. If no manual setting is made, it defaults to the device model + serial number.

- Request URL
  /api/systemctrl/system/getHostname
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)


```
{"result":"ok", "data":{"hostname": "N60-123456"}}
```


- Response Parameter Description（data Field description）

Name	Type	Description
hostname	**[STRING]**	Device Name

1.3.3 Set Device Name

- Interface Description

This interface is used to set the name of the device.

- Request URL
  /api/systemctrl/system/setHostname
- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
name	"N60"	Y	**[STRING]**	Device Name

- Response Parameters (Example)

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

- Response Parameter Description

None

1.3.4 Get HTTP and HTTPS port switch

- Interface Description

This interface is provided to the front-end users to obtain the open/close status of http and https ports on the current device that are saved in the device, and display it on the page.

- Request URL

/api/systemctrl/system/getProtocol

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

None

- Response Parameters (Example)

```
{
    "result": "ok",
    "msg": null,
    "data": {
        "http": true,
        "https": true,
        "httpPort": 80,
        "httpsPort": 443
    }
}
```

- Response Parameter Description

Name	Type	Description
http	**[BOOLEAN]**	Whether to enable HTTP connection
https	**[BOOLEAN]**	Whether to enable HTTPS connection
httpPort	**[INT]**	HTTP port number
httpsPort	**[INT]**	HTTPS port number

1.3.5 Set HTTP and HTTPS port switch

- Interface Description

By default, the machine does not have HTTPS enabled. If you want to use the voice intercom function, you can enable HTTPS through this interface.

- Request URL

/api/systemctrl/system/setProtocol

- Request Method

POST

- Request Parameters (Example)

```
{
"http": true,
"https": true,
"httpPort": 80,
"httpsPort": 443
}
// HTTP and HTTPS cannot be turned off simultaneously.
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
http	true	Y	**[BOOLEAN]**	Whether to enable HTTP connection
https	true	Y	**[BOOLEAN]**	Whether to enable HTTPS connection
httpPort	80	Y	**[INT]**	Set the HTTP port number
httpsPort	443	Y	**[INT]**	Set the HTTPS port number

- Response Parameters (Example)

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

- Response Parameter Description

None

1.3.6 Factory Reset

- Interface Description

By requesting this interface, all settings made on the current device will be cleared, restoring it to a fresh state of the current version.

- Request URL

/api/systemctrl/system/reFactory

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

None

- Response Parameters (Example)

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

- Response Parameter Description

None

1.3.7 Reset

- Interface Description

By requesting this interface, the codec encoding and decoding service can be restarted.

- Request URL

/api/systemctrl/system/restore

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

None

- Response Parameters (Example)

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

- Response Parameter Description
  None

1.3.8 Reboot Device

- Interface Description

By requesting this interface, the device will immediately restart.

- Request URL
  /api/systemctrl/system/reboot
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

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

- Response Parameter Description
  None

1.3.9 Get Network Card Speed

- Interface Description

The current device has two network cards, eth0 and eth1. When requesting this interface, you can specify which network card's speed to return. If no parameter is given, the speed of both network cards will be returned to the user by default.

- Request URL
  /api/systemctrl/system/getNetworkSpeed
  example：
  /api/systemctrl/system/getNetworkSpeed?ifname=eth0
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description

Name	Parameter Value	Value Required	Type	Parameter Description
ifname	"eth0"/"eth1"	N	**[STRING]**	When this parameter is provided in the request, the corresponding network card speed will be returned based on the current parameter. When not filled, both eth0 and eth1 will be returned.

- Response Parameters (Example)

```
{
    "result": "ok",
    "data": {
        "eth0": {
            "receive": 18.55,
            "transmit": 433.52,
            "speed": 1000
```

```
        },
        "eth1": {}
    }
}
```

- Response Parameter Description

Name	Type	Description
receive	**[NUMBER]**	Data Received Between Two Interface Calls
transmit	**[NUMBER]**	Data Sent
speed	**[NUMBER]**	Network Card Speed

1.3.10 Get System Information

- Interface Description

By providing the given parameters, you can obtain specific information. The current version of the device allows selection to enable or disable device firmware version, CPU usage, memory usage, disk usage, and persistent time.

- Request URL

  /api/systemctrl/system/getSystemInfo

  example：

  /api/systemctrl/system/getSystemInfo?version=true&cpu=true&memory=true&disk=true&persisTime=true

- Request Method

  GET

- Request Parameters (Example)

  None

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
version	true	N	**[BOOLEAN]**	Firmware version of the current device
cpu	true	N	**[BOOLEAN]**	Get CPU usage
memory	true	N	**[BOOLEAN]**	Get memory usage
disk	true	N	**[BOOLEAN]**	Get disk usage
persisTime	true	N	**[BOOLEAN]**	Get persistent time usage

- Response Parameters (Example)

```
{
        "result": "ok",
        "data": {
            "version": {
                "ndiVersion": "5.1.3",
                "hardwareVersion": "1.0",
                "softwareVersion": "1.0.1234",
                "serialNumber": "00000000",
                "product":"",
                "oem":""
                },
            "cpu": {
                "precent": 2.624
                },
            "mem": {
                "precent": 16.282,
                "total": 4039368,
                "used": 248088
                },
            "disk": {
```

```
                    "precent": 28,
                    "free": 1350944,
                    "used": 513328,
                    "total": 1983056
                    },
                "persisTime": {
                    "uptime": 1933.73
                    }
                }
            }
```


- Response Parameter Description（data Field description）

Name	Type	Description
version	**[OBJECT]**	Version information, software version, hardware version, NDI version, serial number
cpu	**[OBJECT]**	CPU utilization (%)
memory	**[OBJECT]**	Memory usage percentage (%), total memory (kb), used memory (kb)
disk	**[OBJECT]**	Disk utilization percentage (%), total disk space (kb), free disk space (kb), used disk space (kb)
persisTime	**[OBJECT]**	System uptime (s)

1.3.11 Set device serial number

- Interface Description

The serial number is assigned to the device during manufacturing and can be modified using this interface.

- Request URL
  /api/systemctrl/system/Rpj1Tn92uH3UiCCLPQeyLMd73hElBj6G/setSerial
  example:
  /api/systemctrl/system/Rpj1Tn92uH3UiCCLPQeyLMd73hElBj6G/setSerial?serial_number=7116110002301
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
serial_number	"7116110002301"	Y	**[STRING]**	Device Serial Number

- Response Parameters (Example)


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


- Response Parameter Description
  None

1.3.12 Upload firmware upgrade

- Interface Description

This interface is used to upload firmware upgrades. When using this interface independently, file parameters need to be constructed to proceed.

- Request URL
  /api/systemctrl/system/upload
- Request Method
  POST

- Request Parameters (Example)
  None
- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
file	-	Y	multipart/form-data	Bin File

- Response Parameters (Example)

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

- Response Parameter Description
  None

1.3.13 Switch for testing page service

- Interface Description

This interface is provided to the webui-production page to enable the switch for py-production-module.

- Request URL
  /api/systemctrl/system/production_off
- Request Method
  POST
- Request Parameters (Example)

```
{"value": true}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
value	true	Y	**[BOOLEAN]**	Switch for product testing service. This indicates the setting switch.

- Response Parameters (Example)

```
{"result": "ok", "data": {"value": true}}
```

- Response Parameter Description（data 字段说明）

Name	Parameter Value Example	Value Required	Type	Parameter Description
value	true	Y	**[BOOLEAN]**	Switch for product testing service. This reflects whether the switch is active.

1.3.14 Status of the testing page service

- Interface Description

Used to obtain the switch status of the production service

- Request URL
  /api/systemctrl/system/production_stat
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

{"result": "ok", "data": {"value": true}}

- Response Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
value	true	Y	**[BOOLEAN]**	Status of the product testing service module. Returns whether the current service is active.

2 Video encoding and decoding

2.1 Decoding

2.1.1 Selecting the current decoding stream

- Interface Description

There are two ways to select the decoding stream in decoding mode. The first way is to directly select the current decoding stream from the discovery list, and the second way is to select a preset decoding stream. This API function corresponds to the first way.

- Request URL

  /api/codec/decode/addSpec

- Request Method

  POST

- Request Parameters (Example)

{
"id": 2,
"name": "N60 (Decoding Channel)",
"url": "192.168.42.249:5961",
"group": "public"
}

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
id	2	N	**[INT]**	This parameter is used to pass the index of the current NDI Source in the NDI Source list. It is temporarily not required, but serves as a backup.
name	"N60 (Decoding Channel)"	Y	**[STRING]**	NDI Source Name: The original name of the NDI Source.
url	"192.168.42.249:5961"	Y	**[STRING]**	NDI Source URL: Please note that the current URL format is <IP>:<port>, but it is recommended not to assume that this is always the case. The NDI SDK does not guarantee this. When using it, please keep this URL format and do not attempt to change it.
group	"public"	Y	**[STRING]**	NDI Group where the NDI Source is located. If the Group you pass is an empty string "", according to NDI rules, it represents the default public group.

- Response Parameters (Example)

{"result": "ok"}

- Response Parameter Description
  None

2.1.2 Selecting a decoding stream from the preset

- Interface Description

There are two ways to select the decoding stream in decoding mode. The first way is to directly select the current decoding stream from the discovery list, and the second way is to select a preset decoding stream. This API function corresponds to the second way.

- Request URL
  /api/codec/decode/add
- Request Method
  POST
- Request Parameters (Example)

```
{"id":2}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
id	2	Y	**[INT]**	Specify the Preset ID (0~9) to switch the NDI Source saved in the Preset ID to the current decoding output. If you specify id = "0", it means the current output is not decoded and outputs a blank screen.

- Response Parameters (Example)

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

- Response Parameter Description
  None

2.1.3 Get decoding information

- Interface Description

This interface returns all the information displayed on the decoding page based on the current decoding source conditions through a GET request. This includes the source resolution, frame rate, number of channels for the current decoding source, the number of channels set for the decoding output, and the mapping between output channels and source channels.

- Request URL
  /api/codec/decode/get
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
{
    "result": "ok",
    "data": {
        "name": "DESKTOP-R6TEOQE (vMix - Output 1)",
```

```
"ip": "192.168.38.238",
"online": true,
"resolution": "1920x1080p 60Hz",
"codec": "SHQ2",
"bitrate": 132056,
"deInterlace": false,
"interlaced": 0,
"frame_rate": 60,
"audio": "48.0KHz / 2CH",
"url": "192.168.38.238:5961",
"xRes": 1920,
"yRes": 1080,
"audio_sampling": 48000,
"audio_channels": 2,
"hdcp": 1,
"out_colorspace": 0,
"supported": true,
"output_resolution_choose": "1920x1080",
"output_resolution": "1920x1080p 60Hz",
"hdmi_channels": 2,
"hdmi_channels_mapping": [
    {
        "output_channel": 1,
        "source_channel": 1
    },
    {
        "output_channel": 2,
        "source_channel": 2
    },
    {
        "output_channel": 3,
        "source_channel": 3
    },
    {
        "output_channel": 4,
        "source_channel": 4
    },
    {
        "output_channel": 5,
        "source_channel": 5
    },
    {
        "output_channel": 6,
        "source_channel": 6
    },
    {
        "output_channel": 7,
        "source_channel": 7
    },
    {
        "output_channel": 8,
        "source_channel": 8
    }
],
```

```
            "line_out_channels": 2,
            "line_out_channels_mapping": [
                {
                    "output_channel": 1,
                    "source_channel": 1
                },
                {
                    "output_channel": 2,
                    "source_channel": 2
                },
                {
                    "output_channel": 3,
                    "source_channel": 3
                },
                {
                    "output_channel": 4,
                    "source_channel": 4
                },
                {
                    "output_channel": 5,
                    "source_channel": 5
                },
                {
                    "output_channel": 6,
                    "source_channel": 6
                },
                {
                    "output_channel": 7,
                    "source_channel": 7
                },
                {
                    "output_channel": 8,
                    "source_channel": 8
                }
            ],
            "ndi_connection": "tcp"
        }
}
```


- Response Parameter Description

Name	Type	Description
name	**[STRING]**	Currently decoding NDI Source Name. If currently outputting Preset 0 (Blank), name = "(Blank)".
ip	**[STRING]**	IP address of the currently decoding NDI Source. If there is no decoding, ip = "0.0.0.0".
online	**[BOOLEAN]**	true: The current NDI Source is online; false: The current NDI Source is offline.
resolution	**[STRING]**	The video resolution of the current NDI Source in user-friendly format. If you need to obtain user-friendly information about the video, such as width/height, please refer to the fields “xRes”, “yRes”, “frame_rate”, “interlaced”, etc.
codec	**[STRING]**	The encoding format of the current NDI Source. If the source is offline, it will display "UNKNOWN".

bitrate	**[INT]**	The currently calculated bitrate in Kbps.
deInterlace	**[BOOLEAN]**	Whether the current NDI Source has deinterlace. **Note:** In the current version, this field is set to not deinterlace by default.
interlaced	**[INT]**	The interlace mode of the current NDI Source. If it is "p", the value is 0; if it is "i", the value is 1.
frame_rate	**[INT]**	The original frame rate of the current NDI Source.
audio	**[STRING]**	The original audio sampling rate and number of channels of the current NDI Source.
url	**[STRING]**	The original URL of the currently decoding NDI Source. If there is no decoding NDI Source, url = "" (empty string).
xRes	**[INT]**	The width of the video resolution of the current NDI Source.
yRes	**[INT]**	The height of the video resolution of the current NDI Source.
audio_sampling	**[INT]**	The sampling rate of the audio output after decoding. (This field has no practical effect in the current version and will be used to return the sampling rate of the output in future versions.)
audio_channels	**[INT]**	The sampling rate of the audio output after decoding. (This field will be replaced by the number of HDMI and line-out channels and will be deprecated.)
hdcp	**[INT]**	Whether to enable HDCP (High-bandwidth Digital Content Protection). 1 means enabled, 0 means disabled.
out_colorspace	**[INT]**	The color space type of the output after decoding. 0: RGB444 1: YUV422 2: YUV444
supported	**[BOOLEAN]**	Whether the NDI Source is supported for decoding by the device.
output_resolution_choose	**[STRING]**	The decoding video resolution set in the current system.
output_resolution	**[STRING]**	The actual output video resolution after decoding.
hdmi_channels	**[INT]**	The number of audio channels output from HDMI after decoding.
hdmi_channels_mapping	**[OBJECT]**	The mapping relationship between the HDMI audio channels and the NDI Source audio channel numbers after decoding. **Note:** The channel numbers are represented by Arabic numerals. The basic format is: [{"output_channel": 1, "source_channel": 1}] "output_channel" is the output channel number; "source_channel" is the NDI Source audio channel number.
line_out_channels	**[INT]**	The number of audio channels output from line-out after decoding.
line_out_channels_mapping	**[OBJECT]**	The mapping relationship between the line-out audio channels and the NDI Source audio channel numbers after decoding. **Note:** The channel numbers are represented by Arabic numerals. The basic format is:

		[{"output_channel": 1, "source_channel": 1}] "output_channel" is the output channel number; "source_channel" is the NDI Source audio channel number.
ndi_connection	**[STRING]**	The connection method of the NDI Source.

2.1.4 Setting the output decoding information

- Interface Description

This interface is specifically used to set the output decoding information. The current version includes settings for resolution, HDMI channel count and mapping, line-out channel count and mapping.

- Request URL
  /api/codec/decode/output_set
- Request Method
  POST
- Request Parameters (Example)

```
{
"output_resolution": "1920x1080",
"hdmi_channels": 2,
      "output_framerate": 0,
"hdmi_channels_mapping": [{
"output_channel": 1,
"source_channel": 1
}, {
"output_channel": 2,
"source_channel": 2
}, {
"output_channel": 3,
"source_channel": 3
}, {
"output_channel": 4,
"source_channel": 4
}, {
"output_channel": 5,
"source_channel": 5
}, {
"output_channel": 6,
"source_channel": 6
}, {
"output_channel": 7,
"source_channel": 7
}, {
"output_channel": 8,
"source_channel": 8
}],
"line_out_channels": 2,
"line_out_channels_mapping": [{
"output_channel": 1,
"source_channel": 1
}, {
"output_channel": 2,
"source_channel": 2
}, {
```

```
"output_channel": 3,
"source_channel": 3
}, {
"output_channel": 4,
"source_channel": 4
}, {
"output_channel": 5,
"source_channel": 5
}, {
"output_channel": 6,
"source_channel": 6
}, {
"output_channel": 7,
"source_channel": 7
}, {
"output_channel": 8,
"source_channel": 8
}],
"hdcp": 1,
"out_colorspace": 0
}
```


- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
output_resolution	"1920x1080"	Y	**[STRING]**	Output resolution auto/1280x720/1920x1080/3840x2160
hdmi_channels	2	Y	**[INT]**	HDMI channel count: 2/4/6/8
line_out_channels	2	Y	**[INT]**	LINE OUT channel count
output_framerate	0	Y	**[INT]**	Output frame rate setting (**Note:** This parameter has no practical use in the current version)
hdmi_channels_mapping	[{"output_channel":1,"source_channel":1},{"output_channel":2,"source_channel":2},{"output_channel":3,"source_channel":3},{"output_channel":4,"source_channel":4},{"output_channel":5,"source_channel":5},{"output_channel":6,"source_channel":6},{"output_channel":7,"source_channel":7},{"output_channel":8,"source_channel":8}]	Y	**[OBJECT]**	HDMI channel mapping **Note:** When passing this field to the current interface, be sure to provide values for all 8 channels. The specified channels should be set according to the specified values, while the unspecified channels should be assigned default values in the default order.
line_out_channels_mapping	[{"output_channel":1,"source_channel":1},{"output_channel":	Y	**[OBJECT]**	LINE OUT channel mapping **Note:** When passing this

	2,"source_channel":2}, {"output_channel": 3,"source_channel":3}, {"output_channel": 4,"source_channel":4}, {"output_channel": 5,"source_channel":5}, {"output_channel": 6,"source_channel":6}, {"output_channel": 7,"source_channel":7}, {"output_channel": 8,"source_channel":8}]			field to the current interface, be sure to provide values for all 8 channels. The specified channels should be set according to the specified values, while the unspecified channels should be assigned default values in the default order.
source_channel	1	Y	**[INT]**	Source channel count: Integer ranging from 1 to 8
output_channel	1	Y	**[INT]**	Output channel count
hdcp	1	Y	**[INT]**	Whether to enable High-Bandwidth Digital Content Protection (HDCP). 0 for disabled; 1 for enabled.
out_colorspace	0	Y	**[INT]**	Color space 0: RGB444 1: YUV422 2: YUV444

- Response Parameters (Example)

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

- Response Parameter Description

None

2.1.5 Setting NDI connection mode

- Interface Description

The NDI connection modes currently supported by the device include TCP, RUDP, disabling Multi-TCP, and multicast.

- Request URL

/api/codec/decode/setConnection

- Request Method

POST

- Request Parameters (Example)

```
{"ndi_connection":"rudp"}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
ndi_connection	"rudp"	Y	**[STRING]**	NDI connection mode TCP: tcp RUDP: rudp Disabling Multi-TCP: unicast Multicast: multicast

- Response Parameters (Example)

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

- Response Parameter Description

None

2.2 Discovery Service Module

2.2.1 Set Discovery Server

- Interface Description

This interface connects the device to the discovery server. In encoding mode, the NDI source (this device) will register with the discovery server. In decoding mode, NDI discovery will retrieve the NDI source list from the discovery server. This is suitable for scenarios where NDI is used across network segments or when your network does not support UDP multicast.

- Request URL
  /api/codec/discovery/setDiscoveryServer
- Request Method
  POST
- Request Parameters (Example)
  None
- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
enable	true/false	Y	**[BOOLEAN]**	Set server switch
address		Y	**[STRING]**	Set server address

- Response Parameters (Example)

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

- Response Parameter Description
  None

2.2.2 Get discovery server

- Interface Description

This interface is used to provide the frontend with the currently saved discovery server.

- Request URL
  /api/codec/discovery/getDiscoveryServer
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
{"result":"ok","data":{"enable":false,"address":""}}
```

- Response Parameter Description

Name	Type	Description
enable	**[Boolean]**	Discovery server status
address	**[STRING]**	Discovery server address

2.2.3 Scan and retrieve NDI Source list

- Interface Description

Retrieve the online NDI Source list from the current subnet and manually added addresses on the homepage.

- Request URL
  /api/codec/discovery/scan

- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)


```
{
    "result": "ok",
    "data": [
        {
            "name": "DESKTOP-R6TEOQE (vMix - Output 1)",
            "group": "public",
            "original_url": "192.168.38.238:5961",
            "ip": "192.168.38.238",
            "port": 5961,
            "id": 1,
            "group_name": "public",
            "device_name": "DESKTOP-R6TEOQE",
            "channel_name": "vMix - Output 1",
            "series": "NDI",
            "enable": 1,
            "url": "192.168.38.238:5961"
        }
    ],
    "data_size": 1
}
```


- Response Parameter Description（data field, individual element in the array）

Name	Type	Description
name	**[STRING]**	NDI Source Name: The original name of the NDI Source. If you want to obtain the device name and channel name, please refer to device_name and channel_name.
group	**[STRING]**	NDI Group where the NDI Source is located. If the obtained Group is an empty string "", according to NDI rules, it represents the default public group.
original_url	**[STRING]**	NDI Source URL: Please note that the current URL format is <IP>:<port>, but it is recommended not to take it as a fact. The NDI SDK does not guarantee this. When using it, keep the original value of the URL and do not attempt to change it.
ip	**[STRING]**	IP address
port	**[INT]**	Port number
id	**[INT]**	Index in the discovery list
group_name	**[STRING]**	NDI Group where the NDI Source is located. If the obtained Group is an

		empty string "", according to NDI rules, it represents the default public group.
device_name	**[STRING]**	NDI Device Name parsed from the name, which usually represents the host name of the NDI Source.
channel_name	**[STRING]**	NDI Channel Name parsed from the name
series	**[STRING]**	NDI Source
enable	**[INT]**	1: This preset is defined 0: This preset does not define an NDI Source; in this case, all subsequent fields are meaningless
url	**[STRING]**	NDI Source URL: Please note that the current URL format is <IP>:<port>, but it is recommended not to take it as a fact. The NDI SDK does not guarantee this. When using it, keep the original value of the URL and do not attempt to change it.

2.2.4 Manually add source configuration

- Interface Description

This interface is used to manually add NDI sources, which is a supplement for accessing NDI sources in different subnets due to cross-domain issues.

- Request URL

  /api/codec/discovery/addManualIpsGroups

- Request Method

  POST

- Request Parameters (Example)

```
{
"groups": ["public"],
"manuals": ["128.168.38.207"]
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
groups	["public"]	Y	**[ARRAY(STRING)]**	NDI Group
manuals	["128.168.38.207"]	Y	**[ARRAY(STRING)]**	List of media source addresses

- Response Parameters (Example)

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

- Response Parameter Description

  None

2.2.5 Get manually added NDI source addresses

- Interface Description

This interface is used to provide the frontend with the group list and address list of NDI sources manually configured

on the device.

- Request URL
  /api/codec/discovery/getManualIpsGroups
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

{"result":"ok","data":{"groups":[],"manuals":[]}}

- Response Parameter Description

Name	Type	Description
groups	**[STRING]**	NDI Group where the NDI Source is located. If the obtained Group is an empty string "", according to NDI rules, it represents the default public group.
manuals	**[STRING]**	List of manually added NDI Source addresses.

2.2.6 Setting discovery parameters in decoding mode

- Interface Description

Allows the device to be discovered in decoding mode, and sets the NDI Group and NDI Channel at this time.

- Request URL
  /api/codec/discovery/setDiscoveryGroupChannel
- Request Method
  POST
- Request Parameters (Example)

```
{
    "group_name": "public",
    "channel_name": "Decoding Channel"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
group_name	"public"	Y	**[STRING]**	Sets the NDI Group for decoding discovery, defaulting to "public".
channel_name	"Decoding Channel"	Y	**[STRING]**	Sets the channel name used for identification when discovered.

- Response Parameters (Example)
  None
- Response Parameter Description
  None

2.2.7 Getting the discovery parameters for the device in decoding mode

- Interface Description

Gets the group and channel name of the device discovered in decoding mode.

- Request URL
  /api/codec/discovery/getDiscoveryGroupChannel
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
{
    "result": "ok",
    "data": {
        "group_name": "public",
        "channel_name": "Decoding Channel"
    }
}
```

- Response Parameter Description（data 字段说明）

Name	Type	Description
groups	**[STRING]**	NDI Source Name. This is the original name of the NDI source. If you want to get the device name and channel name, please refer to device_name and channel_name.
channel_name	**[STRING]**	Sets the channel name used for identification when discovered.

2.3 Preset module

Our NDI decoding device has a very useful feature: Presets. The meaning of Preset is that you can add the discovered NDI Source to the 9 Presets represented by 1~9, which are your "favorites". When you need to quickly select the NDI Source added to the Preset for decoding, you only need to specify the ID (1~9) of the Preset.

2.3.1 Get the preset list

- Interface Description

Gets all preset information currently added to the preset in list form.

- Request URL
  /api/codec/preset/get
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
{
    "result": "ok",
    "data": [
        {
```

```
            "id": 1,
            "name": "N60X (Chan1)",
            "url": "192.168.38.61:5961",
            "group": "public",
            "online": "on",
            "warning": "",
            "device_name": "N60X",
            "channel_name": "Chan1",
            "ip": "192.168.38.61",
            "port": 5961,
            "enable": 1
        },
        {
            "id": 2,
            "name": "DESKTOP-R6TEOQE (vMix - Output 1)",
            "url": "192.168.38.238:5961",
            "group": "public",
            "online": "on",
            "warning": "",
            "device_name": "DESKTOP-R6TEOQE",
            "channel_name": "vMix - Output 1",
            "ip": "192.168.38.238",
            "port": 5961,
            "enable": 1
        },
    ]
}
```


- Response Parameter Description

Name	Type	Description
id	**[INT]**	1~9 represents the general NDI Source Preset; 0 represents Blank.
name	**[STRING]**	The name of the NDI Source defined by Preset.
url	**[STRING]**	The original URL of the NDI Source.
group	**[STRING]**	The Group name defined by Preset for the NDI Source.
online	**[STRING]**	Based on the result of network NDI discovery, detects the online status of the NDI Source defined by this Preset: on: The current NDI Source is online (discoverable) off: The current NDI Source is not online (not discoverable) **Please Note:** the online status is determined by the result of NDI Discovery, not whether the NDI Source can be connected and video data obtained. Therefore, it is only for reference and cannot be used as the basis for determining whether the device is online.
warning	**[STRING]**	Used to display whether the NDI Source represented by the current preset is offline.
device_name	**[STRING]**	The Device Name of the NDI Source extracted from the name field.
channel_name	**[STRING]**	The Channel Name of the NDI Source extracted from the name field.
ip	**[STRING]**	The IP address of the NDI Source extracted from the url

		field.
port	**[INT]**	Source port number
enable	**[INT]**	1: This preset is defined. 0: This preset does not define an NDI Source; in this case, all subsequent fields are not meaningful.

2.3.2 Add preset

- Interface Description

By requesting this interface, an effective NDI source can be added to the corresponding position (1~9).

- Request URL

  /api/codec/preset/add

- Request Method

  POST

- Request Parameters (Example)

```
{
    "position": 2,
    "name": "N60X (Chan1)",
    "url": "192.168.38.61:5961",
    "group": "public"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
position	2	Y	**[INT]**	The position where the Preset is added, with a total of 1-9 options.
name	"N60X (Chan1)"	Y	**[STRING]**	The name of the NDI Source, determined by the source.
url	"192.168.38.61:5961"	Y	**[STRING]**	The URL of the NDI Source, determined by the source.
group	"public"	Y	**[STRING]**	The Group of NDI, determined by the source.

- Response Parameters (Example)

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

- Response Parameter Description

  None

2.3.3 Delete preset

- Interface Description

By requesting this interface, the preset in the parameter's corresponding save position can be deleted.

- Request URL

  /api/codec/preset/remove

- Request Method

  POST

- Request Parameters (Example)

```
{"id": 2}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description

id	2	Y	**[INT]**	Specify the Preset ID to be removed, valid range 1~9.

- Response Parameters (Example)

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

- Response Parameter Description

None

2.3.4 Set background color

- Interface Description

This interface sets the display color of the decoding output of the blank page in the preset.

- Request URL

/api/codec/preset/set_blank_color

- Request Method

POST

- Request Parameters (Example)

```
{"BlankColor": "#ffffff"}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
BlankColor	"#ffffff"	Y	**[STRING]**	Specify the Preset ID to be removed, valid range 1~9.

- Response Parameters (Example)

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

- Response Parameter Description

None

2.4 Decoding PTZ camera control

Decoding PTZ camera means that if your device is decoding an NDI source encoded by a PTZ camera, or the PTZ camera is used as a source, connected to an NDI encoding device, and the encoding source of the NDI device is being decoded by your device, you can use the PTZ control panel in the background of the device's decoding mode to control the orientation, focus, zoom, etc. of the PTZ camera on the encoding side.

2.4.1 Decoding mode PTZ control

- Interface Description

PTZ control in decoding mode needs to pass in corresponding values for corresponding parameters.

- Request URL

/api/codec/ptz_decode/ptzControl

- Request Method

GET

- Request Parameters (Example)

```
action:"up"
speed:50
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
action	"up"	Y	**[INT]**	Choose from the following strings: "up", "left", "down", "right".
speed	50	N	**[INT]**	Indicates the speed to be passed in,

				enter a number from 0 to 100, default is 50. For commands such as home, stop-move, stop-all, etc., this value can be left blank.
id	2	N	**[INT]**	Indicates the position passed in, mainly used for commands with preset at the end, enter a number, preferably 1-16.
language	"cn"	N	**[STRING]**	Request header parameter. Currently, only Chinese and English languages are supported in the version, which are "en"/"cn", respectively, for multi-language output of pop-up prompts. **Note:** In the current version, if you want to return the correct error prompt when requesting this interface alone, please be sure to add this parameter to the request header. Otherwise, the msg field in the returned prompt will be null.

- Response Parameters (Example)

```
{"result": "ok"}
{"result": "error", "msg": "Invalid PTZ Motion Parameter"}
{"result": "error", "msg": " PTZ Parameter Setting Incorrect"}
```

- Response Parameter Description

  None

2.5 Encoding

2.5.1 Get the names of all encoding objects

- Interface Description

Encoding objects refer to the objects used internally in the system to control the stream, which are not provided to the frontend and only used in the backend. Currently, there are three types.。

- Request URL

  /api/codec/encoders/list

- Request Method

  POST

- Request Parameters (Example)

  None

- Request Parameter Description

  None

- 响应参数

```
[
    "main",
    "sub",
    "main_full"
]
```

- Response Parameter Description

Name	Type	Description
main	**[STRING]**	NDI|HX main stream object

sub	**[STRING]**	NDI|HX sub-stream object
main_full	**[STRING]**	NDI|HB main and sub-stream common object

2.5.2 Encoding stream switch

- Interface Description

By requesting this interface, you can freely control whether to encode the NDI|HX stream or the NDI|HB stream, or both simultaneously.

- Request URL

  /api/codec/encoders/choose_encode

- Request Method

  POST

- Request Parameters (Example)

```
{"main":1,"main_full":1}
```

- Request Parameter Description

Name	Value	Value Required	Type	Parameter Description
main	1	Y	**[INT]**	NDI|HX stream 0：Off 1：On
main_full	1	Y	**[INT]**	NDI|HB stream 0：Off 1：On

- Response Parameters (Example)

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

- Response Parameter Description

  None

2.5.3 Get encoding stream encoding parameters

- Interface Description

By passing the specified object in {encoder_stream} in the encoding path to this interface, you can obtain the encoding parameter settings for the stream under the current encoding object.

- Request URL

  /api/codec/encoder/{encoder_stream}/get

- Request Method

  POST

- Request Parameters (Example)

  None

- Request Parameter Description

Name	Value	Value Required	Type	Parameter Description
encoder_stream	"main"/"main_full"	Y	**[STRING]**	Path parameters main: NDI|HX main stream sub: NDI|HX sub-stream main_full: NDI|HB main stream **Note:** Under normal circumstances, you can fill in the above three values, but it is better not to fill in "sub" because in the

				current version, the parameters for obtaining the sub-stream value are not accurate.

- Response Parameters (Example)


```
{
    "yuv": "NV12",
    "max_bitrate": 10000,
    "gdr_mode": 0,
    "gop_length": 60,
    "b_frames": 0,
    "target_bitrate": 10000,
    "cpb_size": 1000,
    "initial_delay": 500,
    "periodicity_idr": 60,
    "prefetch_buffer": true,
    "low_bandwidth": false,
    "filler_data": true,
    "control_rate": "cbr",
    "hx_version": 2,
    "mode": "x-h264",
    "profile": "Main",
    "alignment": "nal",
    "mode_version": 1,
    "started": true
}
```


- Response Parameter Description

Name	Type	Description
yuv	**[STRING]**	YUV color encoding format
max_bitrate	**[INT]**	Maximum bit rate of the output video stream
gdr_mode	**[INT]**	Gradual Decoder Refresh (GDR) scheme
gop_length	**[INT]**	Number of frames in a group of consecutive image frames
b_frames	**[INT]**	Setting the number of B-frames
target_bitrate	**[INT]**	Bit rate
cpb_size	**[INT]**	Encoded picture buffer (CPB) specified in the HRD model
initial_delay	**[INT]**	Initial removal delay specified in the HRD model (in milliseconds)
periodicity_idr	**[INT]**	Specifying the periodicity of IDR frames
prefetch_buffer	**[BOOLEAN]**	Enable/disable L2Cache buffer during the encoding process
low_bandwidth	**[BOOLEAN]**	Specify the low bandwidth mode. Reduce the vertical search range used for P-frame motion estimation
filler_data	**[BOOLEAN]**	Enable/disable padding data addition in CBR rate control mode
control_rate	**[STRING]**	Rate control mode. low-latency: low latency; cbr: CBR; vbr: VBR
hx_version	**[INT]**	HX version. 2: HX2; 3: HX3
mode	**[STRING]**	x-h264/x-h265 encoding format

profile	**[STRING]**	Encoding quality level. Available options: Baseline, Extended, Main, High
alignment	**[STRING]**	Delay mode:au/nal
mode_version	**[INT]**	1: x-h264 2: x-h265
started	**[BOOLEAN]**	Stream on/off

2.5.4 Get video source parameters

- Interface Description

By passing the specified object in {encoder_stream} in the encoding path to this interface, you can obtain the basic parameters of the current encoding object at the video source level, including resolution, signal, whether the encoding object is active, and the encoding stream status.

- Request URL
  /api/codec/encoder/{encoder_stream}/get_capture
- Request Method
  POST
- Request Parameters (Example)
  None
- Request Parameter Description

Name	Value	Value Required	Type	Parameter Description
encoder_stream	"main"/"main_full"	Y	**[STRING]**	main: Main stream main_full: FULL NDImain stream

- Response Parameters (Example)

```
{
    "width": 1920,
    "height": 1080,
    "framerate": 60,
    "interlaced": 0,
    "io_mode": 4,
    "device": "/dev/video0",
    "framerate_D": 1000,
    "enable": 1,
    "started": true,
    "signal": "none",
    "resolution": "1920x1080p 60Hz",
    "source": "sdi"
}
```

- Response Parameter Description

Name	Type	Description
width	**[INT]**	Video width
height	**[INT]**	Video height
framerate	**[INT]**	Frame rate
interlaced	**[INT]**	Interlace mode
io_mode	**[INT]**	Video source read mode
device	**[STRING]**	Device node
framerate_D	**[INT]**	Frame rate denominator
enable	**[INT]**	Switch indicating whether the current stream is on or off

started	**[BOOLEAN]**	Switch indicating whether the stream pipeline is open
signal	**[STRING]**	Whether there is audio/video signal
resolution	**[STRING]**	Current format
source	**[STRING]**	Video source access method. sdi, hdmi

2.5.5 Set main stream (NDI|HX) encoding parameters

- Interface Description

By requesting this interface, you can set the encoding parameters for the NDI|HX encoding stream. In the current version, some parameters (such as target_bitrate) take effect in real-time after being set, while some parameters (such as mode_version) will cause the encoding end to restart encoding, resulting in a black screen for a short period.

- Request URL

  /api/codec/encoder/main/set

- Request Method

  POST

- Request Parameters (Example)

```
{
   "target_bitrate":15000,
   "key":"main"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
key	"main"	Y	**[STRING]**	Specify the current stream that needs to be adjusted, main or main_full. **Note:** It needs to match the current URL path.
target_bitrate	100	N	**[INT]**	Encoding quality (bitrate) 55: Very Low 75: Low 100: Default 150: High 200: Very High
mode_version	1	N	**[INT]**	1: x-h264 2: x-h265
hx_version	2	N	**[INT]**	2: NDI | HX2 3: NDI | HX3
yuv	"NV12"	N	**[STRING]**	Color space: NV12: YUV420 8bit NV16: YUV420 10bit NV12_10LE32: YUV422 8bit NV16_10LE32: YUV422 10bit

- Response Parameters (Example)

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

- Response Parameter Description

  None

2.5.6 Set FULL NDI stream (NDI|HB) encoding parameters

- Interface Description

By requesting this interface, you can set the encoding parameters for the NDI|HB encoding stream.

- Request URL

  /api/codec/encoder/main_full/set

- Request Method

  POST

- Request Parameters (Example)

```
{
   "target_bitrate":115,
   "key":"main_full"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
key	"main"	Y	**[STRING]**	Specify the current stream that needs to be adjusted, main or main_full. **Note:** It needs to match the current URL path.
target_bitrate	100	N	**[INT]**	Encoding quality (bitrate) 55: Very Low 75: Low 100: Default 150: High 200: Very High

- Response Parameters (Example)

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

- Response Parameter Description

  None

2.5.7 Get stream information

- Interface Description

By passing the specified object in {encoder_stream} in the encoding path to this interface, you can obtain the stream configuration information for that object.

- Request URL

  /api/codec/streams/{stream_name}/list

- Request Method

  GET

- Request Parameters (Example)

  None

- Request Parameter Description

Name	Value	Value Required	Type	Parameter Description
stream_name	"main"/"main_full"	Y	**[STRING]**	Path parameters

- Response Parameters (Example)

```
// NDI-HX stream
[
    {
        "type": "NDI-HX",
```

```
            "name": "ndi-hx",
            "enable": true,
            "group": "1212222",
            "ttl": "127",
            "channel_name": "Chan1",
            "connection": "disable_rudp",
            "netmask": "255.255.0.0",
            "netprefix": "239.255.0.0",
            "discovery_server": {
                  "enable": false,
                  "address": ""
            },
            "location": null
      }
]

// NDI-HB 流
[
      {
            "group": "",
            "ttl": 127,
            "channel_name": "FULL-24844",
            "connection": "rudp",
            "netmask": "255.255.0.0",
            "netprefix": "239.255.0.0",
            "discovery_server": {
                  "enable": true,
                  "address": "192.168.4.66"
            }
      }
]
```


- Response Parameter Description

Name	Type	Description
type	**[STRING]**	Stream type. NDI-HX, NDI-HB
name	**[STRING]**	Stream name
enable	**[BOOLEAN]**	Whether the current stream is on
group	**[STRING]**	Group name
channel_name	**[STRING]**	NDI channel name
ttl	**[INT]**	Lifetime
connection	**[STRING]**	Connection method
netmask	**[STRING]**	Subnet mask
netprefix	**[STRING]**	Multicast address
discovery_server	**[OBJECT]**	Whether connected to the discovery server. address: Discovery server address enable: Whether connected to the discovery server

2.5.8 Re-stream all streams

- Interface Description

By requesting this interface, both HX and HB streams can be re-streamed. The path parameter can be filled in with "main" or "main_full".

- Request URL
  /api/codec/streams/{stream_name}/reset
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description

Name	Value	Value Required	Type	Parameter Description
stream_name	"main"/"main_full"	Y	**[STRING]**	Path parameters

- Response Parameters (Example)

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

- Response Parameter Description
  None

2.5.9 Get stream-related properties and configurations

- Interface Description

Get the related stream properties and configurations of NDI|HX and NDI|HB.

- Request URL
  /api/codec/streams/{stream_name}/{stream_key}/get
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Path Parameter Explanation

Name	Parameter Value Example	Value Required	Type	Parameter Description
stream_name	"main"	Y	**[STRING]**	Path parameter value main: NDI|HX stream; main_full: NDI|HB stream.
stream_key	"ndi-hx"	Y	**[STRING]**	Path parameter value ndi-hx: This parameter value corresponds to the stream_name parameter value of "main"; ndi-full: This parameter corresponds to the stream_name parameter value of "main_full". **Note:** When the name is "main", the key must be filled with "ndi-hx"; when the name is "main_full", the key must be filled with "ndi-full".

- Response Parameters (Example)

```
{
    "type": "NDI-HX",
    "name": "ndi-hx",
    "enable": true,
    "group": "1212222",
    "ttl": "127",
    "channel_name": "Chan1",
    "connection": "disable_rudp",
    "netmask": "255.255.0.0",
    "netprefix": "239.255.0.0",
```

```
    "discovery_server": {
        "enable": false,
        "address": ""
    },
    "location": null,
    "started": true,
    "source": "sdi"
}
```


- Response Parameter Description

Name	Type	Description
type	**[STRING]**	Stream type. NDI-HX, NDI-HB
name	**[STRING]**	Stream name
enable	**[BOOLEAN]**	Whether the current stream is on
group	**[STRING]**	Group name
channel_name	**[STRING]**	NDI channel name
ttl	**[INT]**	Lifetime
connection	**[STRING]**	Connection method
netmask	**[STRING]**	Subnet mask
netprefix	**[STRING]**	Multicast address
discovery_server	**[OBJECT]**	Whether connected to the discovery server. address: Discovery server address enable: Whether connected to the discovery server
location	**[STRING]**	(Obsolete fields in the current version)
started	**[BOOLEAN]**	Whether the stream is on
source	**[STRING]**	Stream source access method. sdi, hdmi. N60 is fixed as hdmi

2.5.10 Get real-time bitrate and connection count of the stream

- Interface Description

The displayed bitrate on the page is sent from the backend system to the frontend page, while this interface provides an interface for the frontend to actively obtain the current bitrate and connection count.

- Request URL

  /api/codec/streams/{stream_name}/{stream_key}/get_dynamic_state

- Request Method

  GET

- Request Parameters (Example)

  None

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
stream_name	"main"	Y	**[STRING]**	Path parameter value main: NDI|HX stream; main_full: NDI|HB stream.
stream_key	"ndi-hx"	Y	**[STRING]**	Path parameter value ndi-hx: This parameter value corresponds to

				the stream_name parameter value of "main"; ndi-full: This parameter corresponds to the stream_name parameter value of "main_full". **Note:** When the name is "main", the key must be filled with "ndi-hx"; when the name is "main_full", the key must be filled with "ndi-full".

- Response Parameters (Example)

```
{
    "bitrate": 0,
    "connections": 0
}
```

- Response Parameter Description

Name	Type	Description
bitrate	**[INT]**	Real-time bitrate of this stream
discovery_connections	**[INT]**	Real-time connection count of this stream. **Note:**The current device version does not support displaying the real-time connection count. This field is a reserved field and defaults to 0.

2.5.11 Update stream-related configuration

- Interface Description

Through this interface, you can adjust the NDI group, NDI channel, and NDI connection method of the current stream.

- Request URL

  /api/codec/streams/{stream_name}/{stream_key}/set

- Request Method

  POST

- Request Parameters (Example)

```
{
    "group": "",
    "channel_name": "Chan1-248",
"connection": "rudp",
"types": "ndi-hx"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
stream_name	"main"	Y	**[STRING]**	Path parameter value main: NDI|HX stream; main_full: NDI|HB stream.
stream_key	"ndi-hx"	Y	**[STRING]**	Path parameter value ndi-hx: This parameter value corresponds to the stream_name parameter value of "main"; ndi-full: This parameter corresponds to

				the stream_name parameter value of "main_full". **Note:** When the name is "main", the key must be filled with "ndi-hx"; when the name is "main_full", the key must be filled with "ndi-full".
group	"public"	N	**[STRING]**	NDI Group
channel_name	"Chan1-248"	N	**[STRING]**	NDI channel name
connection	"disable_rudp"	N	**[STRING]**	Connection method: disable_rudp: Default rudp: RUDP udp: Disable Multi-TCP multicast: Multicast
types	"ndi-hx"	N	**[STRING]**	Keep consistent with the stream_key parameter value

- Response Parameters (Example)

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

- Response Parameter Description

None

2.6 Mode

The N60 device has the ability to function as both an encoder and a decoder, allowing you to switch between the two.

2.6.1 Set Working Mode (Encoding Mode | Decoding Mode)

- Interface Description

Through this interface, you can freely switch between encoding mode and decoding mode. Please note that after switching, the device will restart and load new resources, which may take around 1 minute.

- Request URL

/api/codec/mode/set?mode=encode

- Request Method

POST

- Request Parameters (Example)

None

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
mode	"encode"/"decode"	Y	**[STRING]**	encode: Encoding mode; decode: Decoding mode

- Response Parameters (Example)

```
true
```

- Response Parameter Description

None

2.6.2 Get Working Mode

- Interface Description

By requesting this interface, you can directly obtain the current encoding/decoding mode of the device.

- Request URL
  /api/codec/mode/get
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
encode
```

- Response Parameter Description

Name	Type	Description
-	**[TEXT]**	encode: Encoding mode; decode: Decoding mode

2.7 Encode PTZ Control

2.7.1 Encode PTZ Control

- Interface Description

By requesting this interface and providing the appropriate actions and speeds, you can simulate controlling the PTZ (Pan-Tilt-Zoom) of the camera through the panel on the webpage. This allows you to control the direction, focus, and zoom of the PTZ.

- Request URL
  /api/codec/ptz/ptzControl
- Request Method
  GET
- Request Parameters (Example)

```
action:"up"
speed:50
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
action	"up"	Y	**[INT]**	Choose from the following strings: "up", "left", "down", "right"
speed	50	N	**[INT]**	Indicate the speed by entering a number from 0 to 100 (default: 50) For commands like "home", "stop-move", "stop-all", etc., this field can be omitted.
id	2	N	**[INT]**	Indicate the position by entering a number, mainly used for commands ending with "preset". It is recommended to use

				numbers from 1 to 16.
language	"cn"	N	**[STRING]**	Request header parameter: Currently, only supports "en" (English) or "cn" (Chinese) for multilingual output in pop-up prompts. **Note:** In the current version, if you want to receive correct error messages when requesting this interface separately, be sure to include this parameter in the request header. Otherwise, the "msg" field in the returned response will be null.

- Response Parameters (Example)

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

- Response Parameter Description

None

2.7.2 Get PTZ Information

- Interface Description

This interface is used to obtain the configuration information for connecting and controlling PTZ cameras on the current device.

- Request URL

/api/codec/ptz/getPtzInfo

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

None

- Response Parameters (Example)

```
{
    "result": "ok",
    "data": {
        "enable": 1,
        "typ": "network",
        "network": {
            "protocol": "TCP",
            "addr": "192.168.38.18",
            "port": 50381,
            "ptz_protocol": "Sony Visca",
            "ptz_addr": "1"
        },
        "serial": {
            "device": "none",
            "baudrate": 9600,
            "parity": "none",
```

```
                "startBits": 8,
                "endBits": 1,
                "xonxoff": 0,
                "rtscts": 0,
                "ptz_protocol": "Sony Visca",
                "ptz_addr": "1"
            }
        }
}
```


- Response Parameter Description

Name	Type	Description
enable	**[BOOLEAN]**	PTZ function status (enabled/disabled)
typ	**[STRING]**	Connection method: network: Network connection serial: Serial connection
network	**[OBJECT]**	For network connection: protocol: Network protocol (TCP/UDP) addr: PTZ camera IP address port: PTZ camera port ptz_protocol: PTZ protocol ("Sony", "Visca", "Pelco-d", or "Pelco-p")
serial	**[OBJECT]**	For serial connection: device: Serial device driver baudrate: Baud rate parity: Parity startBits: Data bits xonxoff: XON/XOFF switch rtscts: RTS/CTS switch

2.7.3 Get PTZ Protocol

- Interface Description

In the current version, the supported protocols for N60 include Sony Visca, Pelco-d, and Pelco-p. By requesting this interface, you can obtain the corresponding protocol names from the "data" field in the response.

- Request URL
  /api/codec/ptz/getPtzProtocol
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)


```
{
    "result": "ok",
    "msg": null,
    "data": [
        "Sony Visca",
        "Pelco-d",
        "Pelco-p"
    ]
}
```

- Response Parameter Description
  None

2.7.4 Get Serial Devices

- Interface Description

If a serial device is connected or a serial cable is directly inserted into the device, you can use this interface to obtain the device node of the serial device.

- Request URL
  /api/codec/ptz/getSerialDevice
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
{
    "result": "ok",
    "data": [
      {
        "device": "ttyUSB0",
        "driver": "FTDI"
      },
    ]
}
```

- Response Parameter Description
  None

2.7.5 Modify PTZ Information

- Interface Description

By providing the corresponding field parameters through this interface, you can modify the PTZ connection and control information saved on the device.

- Request URL
  /api/codec/ptz/modify
- Request Method
  POST
- Request Parameters (Example)

```
{
"enable": 1,
"typ": "network",
"device": "none",
"baudrate": 9600,
"parity": "none",
"startBits": 8,
"endBits": 1,
"xonxoff": 0,
"rtscts": 0,
"ptz_protocol": "Sony Visca",
"ptz_addr": 1,
"protocol": "UDP",
```

"addr": "192.168.38.18",
"port": 50381
}

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
enable	0	Y	**[INT]**	PTZ control switch: 0: Off 1: On
typ	"network"	Y	**[STRING]**	PTZ type: network/serial
device	"ttyUSB0"	N	**[STRING]**	Serial device driver
baudrate	9600	N	**[INT]**	Baud rate
parity		N	**[STRING]**	Parity bit
startBits	8	N	**[INT]**	Data bits
endBits	1	N	**[INT]**	Stop bits
xonxoff	0	N	**[INT]**	XON/XOFF switch
rtscts	0	N	**[INT]**	RTS/CTS switch
ptz_protocol	"Sony Visca"	N	**[STRING]**	PTZ protocol, currently available protocols include: "Sony Visca", "Pelco-d", "Pelco-p"
ptz_addr	1	N	**[INT]**	Serial address location. Fixed parameters within the protocol. Digit, either 1 or 0
protocol	UDP/TCP	N	**[STRING]**	Network protocol
addr	"192.168.38.18"	N	**[STRING]**	PTZ camera IP address
port	50381	N	**[INT]**	PTZ camera port

- Response Parameters (Example)

{"result":"ok","msg":null}

- Response Parameter Description

None

2.8 QR Code Scanning

2.8.1 Virtual User Identity Login

- Interface Description

The first request from the front-end when scanning the QR code, used to generate a virtual user ID responsible for recording the user's login status and unique identification.

- Request URL

/api/codec/qr_code/virtual_login

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

None

- Response Parameters (Example)

```
{
    "result": "ok",
    "data": {
        "current_virtual_user": "1e1ffd2ec06c43e6a9efe6022162ac74"
    }
}
```

- Response Parameter Description

Name	Type	Description
current_virtual_user	**[STRING]**	Identifier ID for virtual login user

2.8.2 Get QR Code ID

- Interface Description

The second request from the front-end when scanning the QR code, used to obtain the unique identification ID of the QR code on the small screen panel for subsequent verification.

- Request URL

  /api/codec/qr_code/qrcode_get

- Request Method

  GET

- Request Parameters (Example)

  None

- Request Parameter Description

  None

- Response Parameters (Example)

```
{
    "result": "ok",
    "data": {
        "qrcode_id": "84598cea1a878c6f2b7e3db89977b0e2"
    }
}
```

- Response Parameter Description

Name	Type	Description
qrcode_id	**[STRING]**	Identifier ID of the QR code

2.8.3 QR Code Verification

- Interface Description

Only one person is allowed to enter the decoding page after scanning the same QR code, but multiple users who have already scanned the code can be online at the same time (refreshing does not affect them).

- Request URL

  /api/codec/qr_code/qrcode_session

- Request Method

  POST

- Request Parameters (Example)

```
{
"current_virtual_user": "1e1ffd2ec06c43e6a9efe6022162ac74",
"qrcode_id": "84598cea1a878c6f2b7e3db89977b0e2",
}
```

- Request Parameter Description

Name	Parameter Value Example	Value	Type	Parameter

		Required		Description
current_virtual_user	"1e1ffd2ec06c43e6a9efe6022162ac74"	Y	**[STRING]**	Virtual user token
qrcode_id	"84598cea1a878c6f2b7e3db89977b0e2"	N	**[INT]**	QR code ID
language	"cn"	N	**[STRING]**	Request header parameter: Currently, only supports "en" (English) or "cn" (Chinese) for multilingual output in error results. **Note:** If no prompt is given, the "msg" field in the returned response will be empty when an error occurs.

- Response Parameters (Example)

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

- Response Parameter Description

None

2.8.4 Check if the Current Mode is Decoding

- Interface Description

This method provides a way for the backend to check the current encoding/decoding mode. It is reserved and currently, the frontend handles the check of the encoding/decoding mode.

- Request URL

/api/codec/qr_code/check_mode

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
language	"en"	N	**[STRING]**	Request header parameter: Currently, only supports "en" (English) or "cn" (Chinese) for multilingual output in error results. **Note:** If no prompt is given, the "msg" field in the returned response will be empty when an error occurs.

- Response Parameters (Example)

Response Result in Decoding Mode：

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

Response Result in Encoding Mode：

```
{
    "result": "verficationError",
    "msg": "Please set the device to decoding mode and re-scan the QR!"
}
```

- Response Parameter Description

None

2.8.5 Get QR Code

- Interface Description

When clicking the QR code icon on the mobile device, it will display the same QR code as the one on the small screen panel.

- Request URL

/api/codec/qr_code/get_qrcode

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

None

- Response Parameters (Example)

```
{
    "result": "ok",
    "data": {
        "image_path":
"192.168.38.217/api/codec/qrcode_img/84598cea1a878c6f2b7e3db89977b0e2_PC.png"
    }
}
```

- Response Parameter Description

Name	Type	Description
image_path	**[STRING]**	Obtain the current QR code image saved on the machine for display on the PC or mobile device.

3 Network

3.1 Network Management

3.1.1 Get All Network Information

- Interface Description

This interface will return the network settings for both network cards.

- Request URL

/api/networkmanager/network/GetLinkinfo

- Request Method

GET

- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)


```
{
    "result": "ok",
    "data": [
        {
            "status": "up",
            "address": "192.168.38.217",
            "netmask": "255.255.255.0",
            "gw": "192.168.38.254",
            "mac": "56:3D:69:9B:5A:C7",
            "method": "static",
            "enable": true,
            "dns": "223.5.5.5;224.5.5.5",
            "device": "eth0"
        },
        {
            "status": "down",
            "address": "0.0.0.0",
            "netmask": "255.255.255.0",
            "gw": "",
            "mac": "44:07:6A:5F:D4:F7",
            "method": "dhcp",
            "enable": true,
            "dns": "223.5.5.5;224.5.5.5",
            "device": "eth1"
        }
    ]
}
```


- Response Parameter Description

Name	Type	Description
status	**[STRING]**	Current network activation status: The status is determined by whether the current network interface is connected to the network. "up": Network activated "down": Network not activated
address	**[STRING]**	IP address, which may be obtained from DHCP or manually set.
netmask	**[STRING]**	Subnet mask
gw	**[STRING]**	Gateway
mac	**[STRING]**	MAC address
method	**[STRING]**	Address acquisition method: dhcp: Automatic DHCP acquisition static: Manual configuration
enable	**[BOOLEAN]**	Current network card validity. In the current version, this field is set to true by default.
dns	**[STRING]**	DNS addresses, separated by semicolons and returned as a string.
device	**[STRING]**	Network Card

3.1.2 Modify current device network information

- Interface Description

Use this interface to modify the network information of the current device. This is a backup interface and is not used to set network information on the actual machine.

- Request URL

  /api/networkmanager/network/ChangeLinkinfo

- Request Method

  POST

- Request Parameters (Example)

```
{
   "ifname": "eth0",
   "cfg": {
                  "address": "0.0.0.0",
                  "netmask": "255.255.255.0",
                  "gw": "",
                  "mac": "44:07:6A:5F:D4:F7",
                  "method": "dhcp",
                  "enable": true,
                  "dns": "223.5.5.5;224.5.5.5",
            }
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
address	"192.168.38.16"	Y	**[STRING]**	Request header parameters: Currently, only Chinese and English languages are supported for displaying error messages in pop-up windows. **Note:** If there is no prompt, the value of the "msg" field will be empty when an error occurs.
netmask	"255.255.255.0"	Y	**[STRING]**	Subnet mask
gw	"192.168.38.254"	Y	**[STRING]**	Gateway
mac	"44:07:6A:5F:D4:F7"	Y	**[STRING]**	MAC address
method	"dhcp"	Y	**[STRING]**	Address acquisition method: dhcp: Automatic DHCP acquisition static: Manual configuration
enable	true	Y	**[BOOLEAN]**	Current network card validity: Manually set
dns	"223.5.5.5;224.5.5.5"	N	**[STRING]**	DNS addresses, separated by semicolons and returned as a string.

- Response Parameters (Example)

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

- Response Parameter Description

  None

3.1.3 Ethernet switch

- Interface Description

Used to control whether the current network card is activated.

- Request URL

  /api/networkmanager/network/SetSwitch

- Request Method

  POST

- Request Parameters (Example)

```
{
   "ifname": "eth0",
   "enable": true
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
ifname	"eth0"	Y	**[STRING]**	Network card
enable	true	Y	**[BOOLEAN]**	Enable/Disable current network card

- Response Parameters (Example)

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

- Response Parameter Description

  None

3.1.4 Get network card information

- Interface Description

Retrieve the corresponding information based on the incoming network card name.

- Request URL

  /api/networkmanager/network/GetEthernets

  example:

  /api/networkmanager/network/GetEthernets?ifname=eth0

- Request Method

  GET

- Request Parameters (Example)

  None

- Request Parameter Description

Name	Value	Value Required	Type	Parameter Description
ifname	"eth0"/"eth1"	Y	**[STRING]**	Network card

- Response Parameters (Example)

```
{
    "result": "ok",
    "data": {
        "status": "up",
        "address": "192.168.38.217",
        "netmask": "255.255.255.0",
        "gw": "192.168.38.254",
        "mac": "56:3D:69:9B:5A:C7",
        "method": "static",
        "enable": true,
```

```
        "dns": "223.5.5.5;224.5.5.5",
        "device": "eth0"
    }
}
```


- Response Parameter Description（data field）

Name	Type	Description
status	**[STRING]**	Current network activation status: The status is determined by whether the current network interface is connected to the network. "up": Network activated "down": Network not activated
address	**[STRING]**	IP address
netmask	**[STRING]**	Subnet mask
gw	**[STRING]**	Gateway
mac	**[STRING]**	MAC address
method	**[STRING]**	Address acquisition method: dhcp: Automatic DHCP acquisition static: Manual configuration
enable	**[BOOLEAN]**	Current network card status. In the current version, this field is set to true by default.
dns	**[STRING]**	DNS addresses, separated by semicolons and returned as a string.
device	**[STRING]**	Network card

3.1.5 Set network card information

- Interface Description

This interface provides the information for setting network cards that are currently in use.

- Request URL

  /api/networkmanager/network/SetEthernets

- Request Method

  POST

- Request Parameters (Example)


```
{
    "ifname": "eth0",
    "address": "192.168.38.217",
    "netmask": "255.255.255.0",
    "gw": "192.168.38.254",
    "mac": "56:3D:69:9B:5A:C7",
    "method": "static",
    "dns": "223.5.5.5;224.5.5.5",
}
```


- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
ifname	"eth0"	Y	**[STRING]**	Network card:"eth0"/"eth1"
address	"192.168.38.217"	Y	**[STRING]**	IP address
netmask	"255.255.255.0"	Y	**[STRING]**	Subnet mask
gw	"192.168.38.254"	Y	**[STRING]**	Gateway
mac	"56:3D:69:9B:5A:C7"	Y	**[STRING]**	MAC address
method	"static"	Y	**[STRING]**	Address acquisition method: dhcp: Automatic DHCP

				acquisition static: Manual configuration
dns	"223.5.5.5;224.5.5.5"	Y	**[STRING]**	DNS server

- Response Parameters (Example)

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

- Response Parameter Description

None

3.1.6 Set link aggregation

- Interface Description

This interface is used to activate and set the link aggregation mode. After setting the link aggregation mode, the normal network card mode will be disabled.

- Request URL

/api/networkmanager/network/SetBonding

- Request Method

POST

- Request Parameters (Example)

```
{
"mac": "96:82:70:FA:36:98",
"address": "192.168.43.77",
"netmask": "255.255.255.0",
"gw": "192.168.43.254",
"dns": ["223.5.5.5", "223.6.6.6", "101.226.4.6", "180.76.76.76", "114.114.114.114", "119.29.29.29"],
"mode": "0",
"method": "dhcp"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
address	"192.168.43.77"	Y	**[STRING]**	Device IP address
netmask	"255.255.255.0"	Y	**[STRING]**	Subnet mask
gw	"192.168.43.254"	Y	**[STRING]**	Gateway
mac	"96:82:70:FA:36:98"	Y	**[STRING]**	MAC address
method	"dhcp"	Y	**[STRING]**	Address acquisition method: dhcp: Automatic DHCP acquisition static: Manual configuration
dns	["223.5.5.5", "223.6.6.6", "101.226.4.6", "180.76.76.76", "114.114.114.114", "119.29.29.29"]	Y	**[STRING]**	DNS server
mode	"0"	Y	**[STRING]**	Aggregation mode: 0: Round-robin load balancing mode 1: Active-passive mode

- Response Parameters (Example)

After a successful request to this interface, the device's IP and other information will be modified, but the interface will not return any values. (Returning a status value is recommended using a sub-thread)

- Response Parameter Description

None

3.1.7 Get link aggregation information

- Interface Description

Request this interface to retrieve link aggregation information, including the current address, gateway, DNS, etc.

- Request URL
  /api/networkmanager/network/GetBondingInfo
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)


```
{
    "status": true,
    "desc": "链路聚合已开",
    "method": "dhcp",
    "mac": "56:3D:69:9B:5A:C7",
    "address": "192.168.38.217",
    "netmask": "255.255.255.0",
    "gw": "192.168.38.254",
    "dns": [
        "223.5.5.5",
        "223.6.6.6",
        "101.226.4.6",
        "180.76.76.76",
        "114.114.114.114",
        "119.29.29.29"
    ],
    "mode": "1",
    "eth0_status": "up",
    "eth1_status": "down"
}
```


- Response Parameter Description

Name	Type	Description
status	**[STRING]**	If the value of this field is false, it means that the current network is in normal mode and not in link aggregation mode. Otherwise, it is in link aggregation mode.
desc	**[STRING]**	IP address
method	**[STRING]**	Address acquisition method: dhcp: Automatic DHCP acquisition static: Manual configuration
mac	**[STRING]**	MAC address
address	**[STRING]**	Device IP address
netmask	**[STRING]**	Subnet mask
gw	**[BOOLEAN]**	Gateway
dns	**[STRING]**	DNS addresses, separated by semicolons and returned as a string.

mode	**[STRING]**	Aggregation mode: 0: Round-robin load balancing mode 1: Active-passive mode
eth0_status	**[STRING]**	Current status of network card 1
eth1_status	**[STRING]**	Current status of network card 2

3.1.8 Restart MDNS

- Interface Description

Requesting this interface will immediately restart MDNS.

- Request URL
  /api/networkmanager/network/ResetMDNS
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

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

- Response Parameter Description
  None

3.2 Static route management

3.2.1 Add Static Route

- Interface Description

Request this interface to add a static route address.

- Request URL
  /api/networkmanager/static/routing/add
- Request Method
  POST
- Request Parameters (Example)

```
{
    "interface":"eth0",
    "target":"192.168.38.12",
    "netmask":"255.255.255.0",
    "gateway":"192.168.38.254",
    "mtu":1500,
    "metric":0
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
interface	"eth0"	Y	**[STRING]**	Network card
target	"192.168.42.76"	Y	**[STRING]**	Destination IP address
netmask	"255.255.255.0"	Y	**[STRING]**	Subnet mask

gateway	"192.168.43.254"	Y	**[STRING]**	Gateway
metric	0	Y	**[INT]**	Route cost
mtu	1500	Y	**[INT]**	Maximum transmission unit

- Response Parameters (Example)

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

- Response Parameter Description

None

3.2.2 Get static route list

- Interface Description

If the device already has a previously saved static route list, you can request this interface to obtain the static route list.

- Request URL

/api/networkmanager/static/routing/get

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

None

- Response Parameters (Example)

```
{
    "interface":"eth0",
    "target":"192.168.38.12",
    "netmask":"255.255.255.0",
    "gateway":"192.168.38.254",
    "mtu":1500,
    "metric":0
}
```

- Response Parameter Description

Name	Type	Description
interface	**[STRING]**	Network card
target	**[STRING]**	Destination
netmask	**[STRING]**	Subnet mask
gateway	**[STRING]**	Gateway
metric	**[INT]**	Route cost
mtu	**[INT]**	Maximum transmission unit
status	**[INT]**	Route enable status

3.2.3 Modify static route

- Interface Description

Request this interface to modify the specified static route.

- Request URL

/api/networkmanager/static/routing/modify

- Request Method

POST

- Request Parameters (Example)

```
{
    "id":1,
    "interface":"eth0",
    "target":"192.168.38.12",
    "netmask":"255.255.255.0",
    "gateway":"192.168.38.254",
    "mtu":1500,
    "metric":0
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
id	1	Y	**[INT]**	Route list ID
interface	"eth0"	Y	**[STRING]**	Network card
target	"192.168.38.12"	Y	**[STRING]**	Destination
netmask	"255.255.255.0"	Y	**[STRING]**	Subnet mask
gateway	"192.168.38.254"	Y	**[STRING]**	Gateway
metric	1500	Y	**[INT]**	Route cost
mtu	0	Y	**[INT]**	Maximum transmission unit

- Response Parameters (Example)

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

- Response Parameter Description

None

3.2.4 Delete static route

- Interface Description

Request this interface to delete a single static route or delete multiple static routes in batches. Refer to the example for the given values.

- Request URL

/api/networkmanager/static/routing/del

- Request Method

POST

- Request Parameters (Example)

```
{
  "ids": 1
}
{
  "ids": [1, 2]
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
ids	1	Y	**[INT]/[ARRAY(INT)]**	Route ID or an array of IDs

- Response Parameters (Example)

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

- Response Parameter Description

None

3.2.5 Static route switch

- Interface Description

Requesting this interface allows you to enable/disable a specified route.

- Request URL
  /api/networkmanager/static/routing/set
- Request Method
  POST
- Request Parameters (Example)

```
{
    "id": 1,
    "status": 1
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
id	1	Y	**[INT]**	Route ID
status	0/1	Y	**[INT]**	Enable/Disable specified route.

- Response Parameters (Example)

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

- Response Parameter Description
  None

4 KiloLink

4.1 KiloLink Client Service

Our NDI devices provide the functionality to access the KiloLink server for centralized management of devices. By enabling and disabling the KiloLink client service on the device, you can manage the KiloLink connections.

4.1.1 Get Current Device Network Interfaces

- Interface Description

This interface is used to retrieve the list of network interfaces on the current device. In normal mode, it will return "eth0" and "eth1", while in link aggregation mode, it will return "bond0".

- Request URL
  /api/KiloLink/networks
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
{
    "result": "ok",
    "list": [
        "bond0"
```

```
    ]
}
```

- Response Parameter Description

4.1.2 Get KiloLink client settings

- Interface Description

This interface is mainly used by the front-end to retrieve the current KiloLink client settings saved on the device.

- Request URL
  /api/KiloLink/get
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
{
    "result": "ok",
    "cfg": {
        "ip": "",
        "port": 50000,
        "ifname": [],
        "key": "",
        "crypto": false,
        "enable": false
    },
    "status": 1
}
```

- Response Parameter Description（cfg Fields）

Name	Type	Description
ip	**[STRING]**	KiloLink server address
port	**[INT]**	Service port
ifname	**[ARRAY(STRING)]**	Network interface
key	**[STRING]**	Authorization code
crypto	**[BOOLEAN]**	Is Encryption enabled
enable	**[BOOLEAN]**	Is KiloLink service enabled

4.1.3 Set KiloLink client configuration parameters

- Interface Description

This interface is used to set the parameters for the KiloLink client in preparation for connecting to the KiloLink server.

- Request URL
  /api/KiloLink/set
- Request Method
  POST
- Request Parameters (Example)

```
{
    "ip": "",
    "port": 50000,
```

```
    "ifname": ["eth0"],
    "key": "",
    "crypto": false,
    "enable": false,
}
```


- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
ip	"192.168.41.211"	Y	**[STRING]**	KiloLink server address
port	5421	Y	**[INT]**	Service port
ifname	["eth0"]	Y	**[ARRAY(STRING)]**	Network interface
key	"is8931ji"	Y	**[STRING]**	Authorization code
crypto	false	Y	**[BOOLEAN]**	Is Encryption enabled
enable	false	Y	**[BOOLEAN]**	Is KiloLink service enabled

- Response Parameters (Example)


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


- Response Parameter Description

  None

4.1.4 Validate token

- Interface Description

This interface is used to validate requests from the KiloLink server for secure verification.

- Request URL

  /api/KiloLink/check_proxy_token?token=xxx

- Request Method

  GET

- Request Parameters (Example)

  None

- Request Parameter Description

  None

- Response Parameters (Example)


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


- Response Parameter Description

  None

5、Voice Intercom

5.1 Talk server

5.1.1 Add Device

- Interface Description

This interface allows manual addition of other devices that can be used for voice intercom on the current device.

- Request URL

  /api/talk/addDevice

- Request Method

POST

- Request Parameters (Example)

```
{
   "type": "device",
   "ip": "192.168.40.100",
   "http_port": 80,
   "https_port": 443
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
type	"device"	Y	**[STRING]**	Type: "device", "server"
ip	"192.168.40.100"	Y	**[STRING]**	Device IP
http_port	80	Y	**[INT]**	HTTP port number
https_port	443	Y	**[INT]**	HTTPS port number

- Response Parameters (Example)

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

- Response Parameter Description

None

5.1.2 Set connection between local device and target device

- Interface Description

This interface is used to establish or disconnect the connection between the local device and the target device. After calling this interface, the internal logic will further initiate a request to the interface (interface 5.1.12) on the target device. If the target device also successfully establishes the communication connection, voice intercom can be conducted.

- Request URL

/api/talk/setLDLinkDEnable

- Request Method

POST

- Request Parameters (Example)

```
{
   "l_dev_ip": "192.168.38.74",
   "dev_ip": "192.168.40.100",
   "enable": true
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
l_dev_ip	"192.168.38.74"	Y	**[STRING]**	Local device IP
dev_ip	"192.168.40.100"	Y	**[STRING]**	Target device IP
enable	true	Y	**[BOOLEAN]**	Call status true: establish communication connection; false: close communication connection

- Response Parameters (Example)

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

- Response Parameter Description

None

5.1.3 Set connection between web page and target device

- Interface Description

The connection between the web page and the target device refers to the direct connection between the web page, i.e., the PC terminal, and the device. After a successful connection, voice communication between the PC terminal and the target device can be conducted.

- Request URL

  /api/talk/setWLinkDEnable

- Request Method

  POST

- Request Parameters (Example)

```
{
   "dev_ip": "192.168.43.176",
   "enable": true
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
dev_ip	"192.168.43.176"	Y	**[STRING]**	Target device IP
enable	true	Y	**[BOOLEAN]**	Call status true: establish communication connection; false: close communication connection

- Response Parameters (Example)

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

- Response Parameter Description

  None

5.1.4 Set connection between device and voice server

- Interface Description

This interface is used to establish a connection between the device and the voice server. After a successful connection, communication can be conducted with all devices currently connected to the voice server.

- Request URL

  /api/talk/setLDLinkSEnable

- Request Method

  POST

- Request Parameters (Example)

```
{
   "l_dev_ip": "192.168.35.11",
   "ser_ip": "192.168.40.100",
   "enable": true
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
l_dev_ip	"192.168.35.11"	Y	**[STRING]**	Local device IP
ser_ip	"192.168.40.100"	Y	**[STRING]**	Target server IP
enable	true	Y	**[BOOLEAN]**	Call status true: establish communication connection; false: close communication connection

- Response Parameters (Example)

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

- Response Parameter Description
  None

5.1.5 Query device connection status

- Interface Description

This interface is used to query whether the current device has established a voice connection.

- Request URL
  /api/talk/queryLDeviceConnectStatus
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
{
    "result": "ok",
    "msg": {
        "connect_status": "noconnect",
        "connect_ip": "0.0.0.0"
    }
}
```

- Response Parameter Description

Name	Type	Description
connect_status	**[STRING]**	Device connection status Not connected: noconnect; Connecting: connecting; Connected: connected
connect_ip	**[STRING]**	Device IP

5.1.6 Query device list

- Interface Description

This interface is used to query all devices and voice servers that can be used for voice intercom, including those obtained through automatic search and manual addition, saved on the device.

- Request URL
  /api/talk/queryDeviceList
- Request Method
  GET
- Request Parameters (Example)
  None
- Request Parameter Description
  None
- Response Parameters (Example)

```
{
    "result": "ok",
    "msg": {
        "local_device": {    // Local device
            "name": "N60-248",      // Device name
            "ip": "192.168.43.248",    // Device IP
```

```
                "type": "device",    // Device type
                "usb_status": false, // USB status. true: USB inserted, false: USB not inserted
                "talk_status": false, // Communication status. true: in communication, false: not in communication
                "connect_status": "noconnect", // Connection status. Not connected: "noconnect" Connecting:
"connecting" Connected: "connected"
                "connect_ip": "0.0.0.0", // Connected device IP
                "connect_type": null,    // Type of connected device. None: no connection "web": web "device":
device "server": server
                "version": null, // Version of local device's communication service
                "http_port": 80,    // Local device's open http port
                "https_port": 443    //Local device's open https port
            },
            "other_device": [    //Other devices
                {
                    "name": "E3-1234567891206",    //Device name
                    "ip": "192.168.43.141",    // Device IP
                    "type": "device",    // Device type. "device" or "server"
                    "discovery_mode": "auto",    // Discovery mode. "auto": automatic discovery, "manual":
manual addition
                    "selected": false,    // Whether it is selected for connection. true: select this device to connect
with the local device, false: not selected
                    "connect_status": "noconnect",    // Connection status. Not connected: "noconnect"
Connecting: "connecting" Connected: "connected"
                    "talk_status": "ready",    //Communication status. "ready": default status when the device is
not connected "talking": in conversation after the device is connected "busy": device is in conversation with other
object
                    "connect_error_info": null, // Connection failure information
                    "version": "1.10.0001",    // Version of device's communication service
                    "http_port": 80,    // Device's open http port
                    "https_port": 443,    // Device's open https port
                    "auth_code": null      // Authorization code. None for device type
                }
            ]
        }
}
```


5.1.7 Delete device

- Interface Description

This interface is used to delete a device by providing its IP address.

- Request URL

  /api/talk/deleteDevice

- Request Method

  POST

- Request Parameters (Example)


```
{
   "device_ip": "192.168.43.176"
}
```


- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description

device_ip	"192.168.43.176"	Y	**[STRING]**	Device IP

- Response Parameters (Example)

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

- Response Parameter Description

None

5.1.8 Query serial number of local device

- Interface Description

This interface is used to retrieve the serial number of the local device.

- Request URL

/api/talk/queryLocalDeviceSerial

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

None

- Response Parameters (Example)

```
{
    "result": "ok",
    "msg": {
        "serial": "012354640124"
    }
}
```

- Response Parameter Description（msg field）

Name	Type	Description
serial	**[STRING]**	Serial number

5.1.9 Talk server version

- Interface Description

This interface is used to retrieve the current version number of the voice intercom server.

- Request URL

/api/talk/talkVersion

- Request Method

GET

- Request Parameters (Example)

None

- Request Parameter Description

None

- Response Parameters (Example)

```
{
    "result": "ok",
    "msg": {
        "version": "1.10.0001"
    }
}
```

- Response Parameter Description

Name	Type	Description
version	**[STRING]**	Version field

5.1.10 Set language

- Interface Description

This interface is used to set the display language of the voice intercom module on the background web page.

- Request URL

  /api/talk/setLanguage

- Request Method

  POST

- Request Parameters (Example)

```
{
   "language": "cn"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
language	"cn"	Y	**[STRING]**	cn: English mode; zh: Chinese mode

- Response Parameters (Example)

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

- Response Parameter Description

  None

5.1.11 Notify device for call deletion (interface not accessible to the front-end)

- Interface Description

This interface is mainly used for the local device to act as a server, receiving requests from the other device to delete the call and subsequently deleting the connection between the two parties.

- Request URL

  /api/talk/notifyDevDelTalk

- Request Method

  POST

- Request Parameters (Example)

```
{
   "c_ip": "192.168.40.10",
   "s_ip": "192.168.40.100"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
c_ip	"192.168.40.10"	Y	**[STRING]**	IP of the other device
s_ip	"192.168.40.100"	Y	**[STRING]**	IP of the local device

- Response Parameters (Example)

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

- Response Parameter Description

  None

5.1.12 Device request connection, device-to-device connection (interface not accessible to the front-end)

- Interface Description

This interface is mainly used for the local device to act as a server, receiving connection requests from the other device and establishing a connection between the two parties.

- Request URL

  /api/talk/dLinkLDEnable

- Request Method

  POST

- Request Parameters (Example)

```
{
   "c_dev_ip": "192.168.40.10",
   "c_dev_info": {
      "version": "1.10.0001",
      "http_port": 80,
      "https_port": 443
   },
   "s_dev_ip": "192.168.40.100"
}
```

- Request Parameter Description

Name	Parameter Value Example	Value Required	Type	Parameter Description
c_dev_ip	"192.168.40.10"	Y	**[STRING]**	IP of the requesting device
c_dev_info	-	-	**[OBJECT]**	Device information of the requesting device
version	"1.10.0001"	Y	**[STRING]**	Version number
http_port	80	Y	**[INT]**	HTTP port number
https_port	443	Y	**[INT]**	HTTPS port number
s_dev_ip	"192.168.40.100"	Y	**[STRING]**	IP of the requested device (local device)

- Response Parameters (Example)

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

- Response Parameter Description

  None