

Yealink SkySound 3rd-Party Control API V1.2

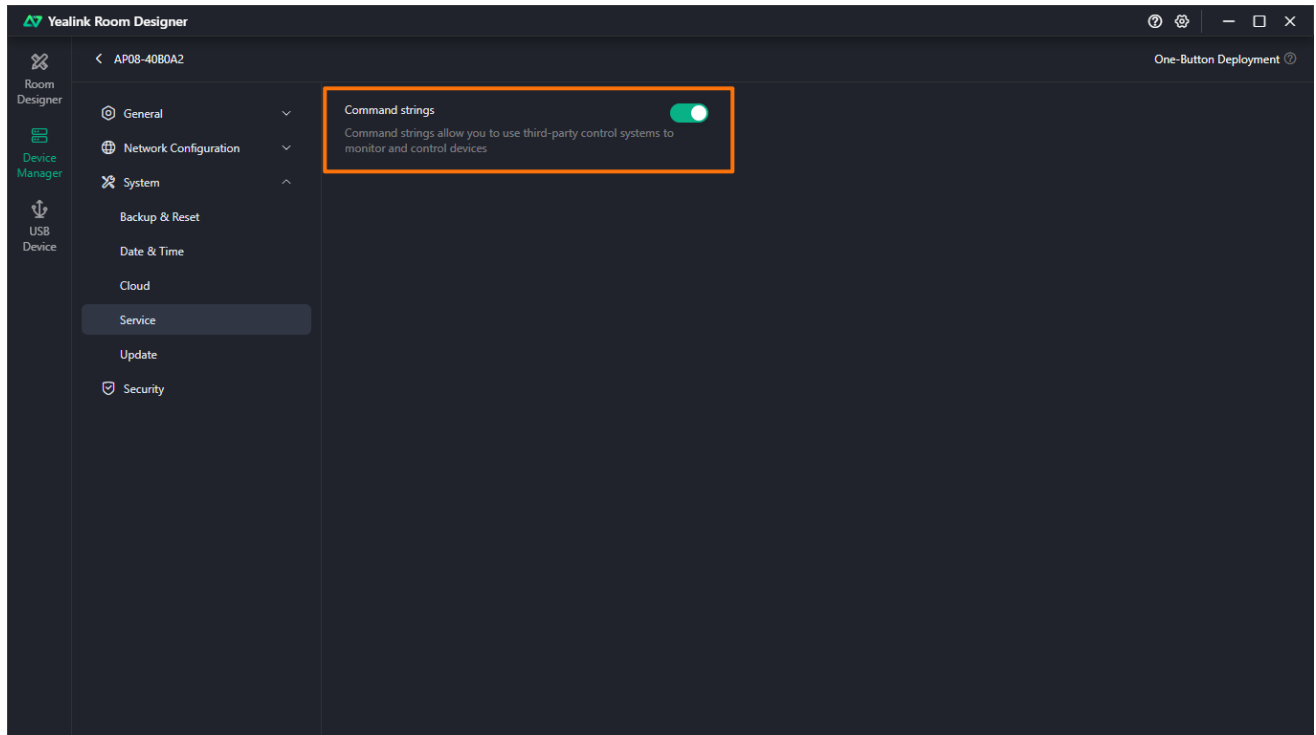
Purpose

This document provides a detailed overview of the functionality and syntax of third-party control API commands for Yealink SkySound devices, with the aim of guiding readers through the integration of control features for SkySound products.

Network Configuration

- This device supports network mode control of the host device, using the TCP protocol to establish connections. Clients need to know the host's IP address and port number to connect to the host.

- The following parameters need to be configured for the network mode:
 - For devices that require control, you need to enable the command string switch on the device. In Yealink Room Designer, go to the **Device Management** page, select the relevant device and open its **Settings** page. Then, under **System** > **Service**, enable the **Command String** option.



- Set the control port for the central controller's network mode to 6024.
- The IP address can be checked as needed.
- If password authentication is enabled on the central controller's network mode, you must enter and verify the correct password over the TCP connection before you can execute control commands. The password input format should be "XXXX\r\n", where XXXX represents the correct password.

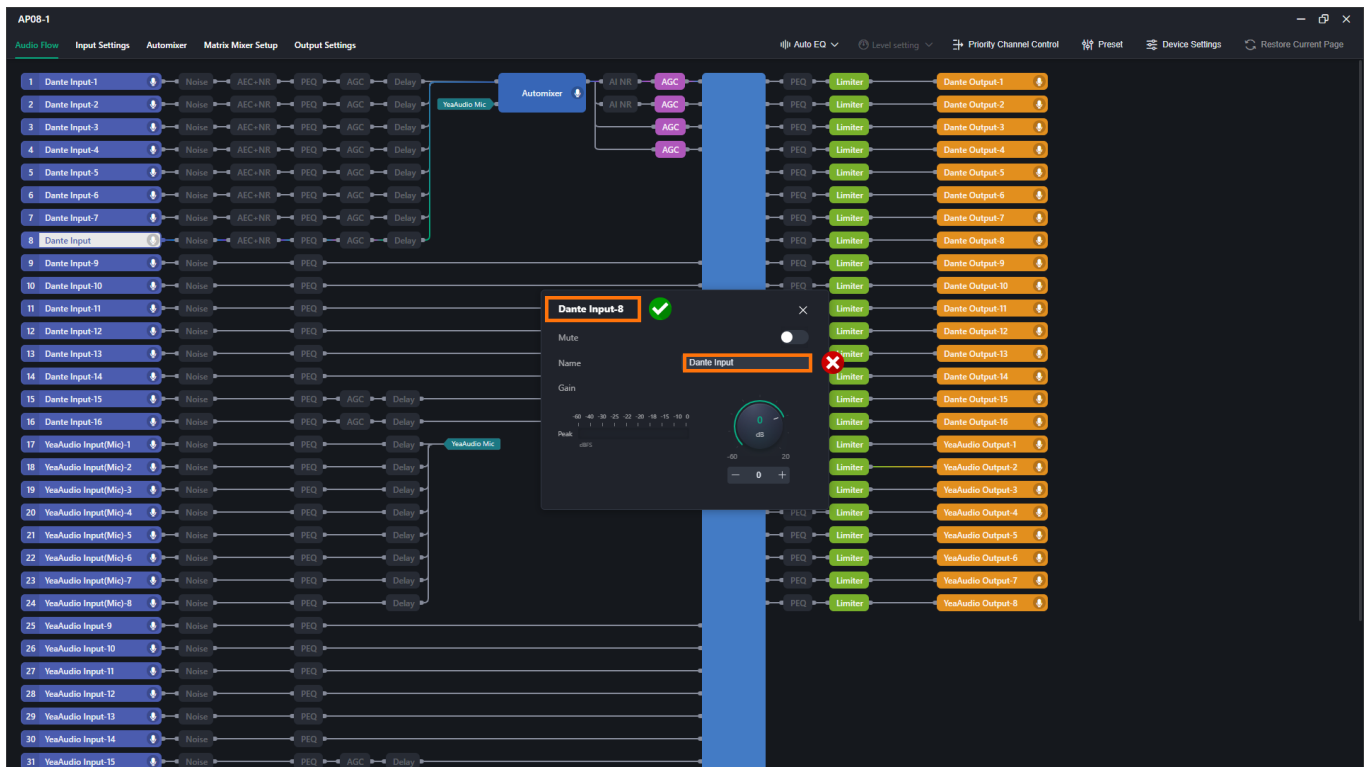
Sample:

- For example, after establishing a TCP connection, receiving the message "Password:\r\n" indicates that authentication credentials are required.
- If the Password you entered is incorrect, the connection will return "Invalid Password!\r\nPassword:\r\n", prompting you to re-enter your Password for authentication.
- As long as you enter the correct password, the connection will return "welcome yealink centralcontrol."

Note:

- If the command you entered returns the same command as output, there may be a formatting or syntax error. Please check that the command format and content are correct.

- When using commands that require a channel name, you must use the actual channel name and not the display name.



API Syntax

The following outlines the syntax of the Yealink SkySound API:

- All command keywords should be in lowercase.
- Basic command structure: "command + parameters".
- Parameter format: parameter name followed by parameter value, separated by a colon (:). The overall format is transmitted as a JSON string.

Packet Format

Message Format Instructions: Use `\r\n` as the end delimiter. All API commands must include `\r\n` as the end delimiter.

Response Format Description: `\r\n` serves as the end delimiter. All response packets are terminated with `\r\n` as the end delimiter.

Basic command syntax:

- command + `\r\n` (no parameter type)
- Command + Space + JSON string + `\r\n` (with parameter type)
- Command + Space + app:value + Space + JSON string + `\r\n` (with parameters and app parameter type)

API Command Introduction

The currently supported API commands are listed below. It is recommended that readers refer to the command documentation and use TCP debugging tools during troubleshooting to build and test API commands based on the supported environments described above. Supported commands and parameters may vary depending on the version and model.

Optional parameters in API command arguments are enclosed in square brackets [].

Get System Hardware Information

- Return Parameters

Name	Type	Description
model	string	Device model
firmware	string	Software version number
hardware	string	Hardware version number
serialnumber	string	Device serial number
macaddress	string	Device mac address
cc-version	string	Central Control version number
vendor	string	Device manufacturer name

- Description

System command	System get sys-info
Supported Model	AP08; CM20; CM50; CS10; CS10-D; AVBridge
Request Format	system get sys-info\r\n
Return Format	system get sys-info { "model": "AP08", "firmware": "294.435.0.20", "hardware": "294.0.0.0.0.0", "serialnumber": "203097G070000041", "macaddress": "44:db:d2:56:b9:56", "cc-version": "10.0.251.4", "vendor": "Yealink" }
Sample	Send: system get sys-info Return: system get sys-info { "model": "AP08", "firmware": "294.435.0.20", "hardware": "294.0.0.0.0.0", "serialnumber": "203097G070000041", "macaddress": "44:db:d2:56:b9:56", "cc-version": "10.0.251.4", "vendor": "Yealink" }

Get Network Information

- Return Parameters

Name	Type	Value Range	Note
network-list	network_info[]	None	Network information list

network_info

Name	Type	Value Range	Note
type	int	[0,1]	Get network method 0: Dynamic 1: Static
port-type	int	[0,1,2]	Network port type 0: Wired Port 1 1: Wired Port 2 2: Wireless Port 3: AP Port
mode	int	[0,1,2]	IP type 0: IPv4 1: IPv6 2: IPv4 and IPv6
ip	string	None	IP address
mask	string	None	Subnet mask
gateway	string	None	Gateway
primary-dns	string	None	Preferred dns server name
second-dns	string	None	Alternative dns server name

- Description

Network command	Network get info
Supported Model	AP08; CM20; CM50; CS10; CS10-D; AVBridge
Request Format	network get info\r\n
Return Format	network get info {"network-list":[{"type":0,"port-type":2,"mode":0,"ip":"","mask":"","gateway":"","primary-dns":"","second-dns":""}, {"type":0,"port-type":0,"mode":0,"ip":"10.50.67.59","mask":"255.255.255.0","gateway":"10.50.67.254","primary-dns":"10.100.1.10","second-dns":"192.168.1.22"}]}\r\n

Sample	Send: network get info \r\n Return: network get info {"network-list":[{"type":0,"port-type":2,"mode":0,"ip":"","mask":"","gateway":"","primary-dns":"","second-dns":""}, {"type":0,"port-type":0,"mode":0,"ip":"10.50.67.59","mask":"255.255.255.0","gateway":"10.50.67.254","primary-dns":"10.100.1.10","second-dns":"192.168.1.22"}]} \r\n
--------	---

Set the Device Location Switch

• Request Parameters

Name	Type	Value Range	Necessity	Description
status	string	{on,off}	Yes	on: Enable device location off: Disable device location

• Description

System command	System set locating-status
Supported Model	AP08; AVBridge; CM20; CM50; CS10; CS10-D
Request Format	system set locating-status {"status":"on"}\r\n
Return Format	system set locating-status {"status":"on"}\r\n
Sample	Send: system set locating-status {"status":"on"}\r\n Return: system set locating-status {"status":"on"}\r\n

Set System Status

• Request Parameters

Name	Type	Value Range	Necessity	Description
status	string	[reboot,reset]	Yes	System status reboot: Restart reset: Factory reset

• Description

System command	System set status {"status":"<value>"}
Supported Model	AP08; AVBridge; CM20; CM50; CS10; CS10-D
Request Format	system set status {"status":"reboot",["sn":"506607D117000009"]} \r\n
Return Format	system set status {"status":"reboot",["sn":"506607D117000009"]} \r\n

Sample	Send: system set status {"status":"reboot"} \r\n Return: system set status {"status":"reboot"} \r\n
--------	--

Get LED Light Switch Status

• Return Parameters

Name	Type	Value Range	Description
enable	string	{on,off}	on: Enabled off: Disabled

• Description

System command	System get led-preset enable
Supported Model	CM20; CM50; CS10; CS10-D
Request Format	system get led-preset enable\r\n
Return Format	system get led-preset enable {"enable":"off"}\r\n
Sample	Send: system get led-preset enable\r\n Return: system get led-preset enable {"enable":"off"}\r\n

Set LED On/Off

• Request Parameters

Name	Type	Value Range	Necessity	Description
enable	string	{on, off}	Yes	on: Turn on the LED light off: Turn off the LED light Note: Both turning on and turning off refer to the LED light for the corresponding event.

• Description

System command	System set led-preset-status
Supported Model	CM20; CM50; CS10; CS10-D
Request Format	system set led-preset enable {"enable":"on"}\r\n
Return Format	system set led-preset enable {"enable":"on"}\r\n
Sample	Send: system set led-preset enable {"enable":"on"}\r\n Return: system set led-preset enable {"enable":"on"}\r\n

Get the Color of the LED Event

• Request Parameters

Name	Type	Value Range	Necessity	Description
event	string	{mute, unmute}	Yes	Event for retrieving the color of the light

• Return Parameters

Name	Static NAT/Type	Value Range	Description
event	string	{mute, unmute}	-
color	string	{red, green, orange, #xxxxxx}	Light color

• Description

System syntax	System get led-preset config
Supported Model	CM20; CM50
Request Format	system get led-preset config {"event":"unmute"}\r\n
Return Format	system get led-preset config {"event":"unmute","color":"#856974"}\r\n
Sample	Send: system get led-preset config {"event":"unmute"}\r\n Return: system get led-preset config {"event":"unmute","color":"#856974"}\r\n

Set the LED Event Color

• Request Parameters

Name	Type	Value Range	Necessity	Description
event	string	{mute,unmute}	Yes	LED event
color	string	{red, green,orange, #xxxxxx}	Yes	The color of the indicator light for the corresponding event

• Description

System command	System set led-preset config
Supported Model	CM20; CM50
Request Format	system set led-preset config {"event":"mute","color":"red"}\r\n
Return Format	system set led-preset config {"event":"mute","color":"red"}\r\n
Sample	Send: system set led-preset config {"event":"mute","color":"red"}\r\n Return: system set led-preset config {"event":"mute","color":"red"}\r\n

Get Device Channel Audio Source Information

- **Return Parameters**

Name	Type	Value Range	Necessity	Description
max_channel_count	int	-	Yes	Maximum number of channels
channel_enable	bool[]	-	Yes	Channel usage status

- **Description**

Audio command	Audio get lobe-enable
Supported Model	CM50
Request Format	audio get lobe-enable\r\n
Return Format	audio get lobe-enable {"max_channel_count":8,"channel_enable": [true,true,true,true,true,true,true,true]}\r\n
Sample	Send: audio get lobe-enable\r\n Return: audio get lobe-enable {"max_channel_count":8,"channel_enable": [true,true,true,true,true,true,true,true]}\r\n

Get Device Audio Channel Activation Status

- **Return Parameters**

Name	Type	Value Range	Necessity	Description
active_status	bool[]	None	Yes	Channel activation status

- **Description**

Audio command	Audio get lobe-activation
Supported Model	CM50
Request Format	audio get lobe-activation\r\n
Return Format	audio get lobe-activation {"active_status":[true,true,true,true,true,true,true,true]}\r\n
Sample	Send: audio get lobe-activation\r\n Return: audio get lobe-activation {"active_status":[true,true,true,true,true,true,true,true]}\r\n

Get Speaker's Location Coordinates

- Return Parameters

Name	Type	Value Range	Necessity	Description
doa	doa[]	None	Yes	Device doa array information

doa

Name	Type	Value Range	Necessity	Description
dev_type	string	None	Yes	Device Type
dev_sn	string	None	Yes	Device serial number
snd_src_num	int	None	Yes	Number of sound sources
snd_src	snd_src[]	None	Yes	Audio source data

snd_src

Name	Type	Value Range	Necessity	Description
x	int	None	Yes	X-axis value
y	int	None	Yes	Y-axis value
z	int	None	Yes	Z-axis value

- Description

Audio command	Audio get sound-locate
Supported Model	CM50
Request Format	audio get sound-locate\r\n
Return Format	audio get sound-locate {"doa": [{"dev_type": "CM50", "dev_sn": "sf sf sf sf", "snd_src_num": 1, "snd_src": [{"x": 0, "y": 0, "z": 0}]}]}\r\n
Sample	Send: audio get sound-locate\r\n Return: audio get sound-locate {"doa": [{"dev_type": "CM50", "dev_sn": "sf sf sf sf", "snd_src_num": 1, "snd_src": [{"x": 0, "y": 0, "z": 0}]}]}\r\n

Get Audio Parameter Presets

- Request Parameters

Name	Type	Required	Constraint	Chinese Name	Description
------	------	----------	------------	--------------	-------------

preset_info	[object]	true	none	Preset information	none
preset_type	string	true	none	Preset Type	none
preset_id	string	true	none	Preset ID	Default unique ID
preset_name	string	true	none	Preset Name	When preset_type is set to "official preset", the name and id are identical.

- **Description**

Audio command	Audio get preset
Supported Model	AP08; AVBridge; CM50; CM20
Request Format	audio get preset\r\n
Return Format	audio get preset {"preset_info":{"preset_type":"<value>","preset_name":"<value>","preset_id":"<value>"} }\r\n
Sample	Send: audio get preset\r\n Return: audio get preset {"preset_info": [{"preset_type":"custom_preset","preset_name":"Custom1","preset_id":"OFQjdE"}, {"preset_type":"official_preset","preset_name":"general_preset","preset_id":"general_preset"}] }\r\n

Set Audio Parameter Presets

- **Request Parameters**

Name	Type	Value Range	Necessity	Description
preset_name	string	/	Yes	Preset unique name, use UTF-8 for Chinese

- **Description**

Audio command	Audio set preset {"preset_name":"<value>"}
Supported Model	AP08; AVBridge; CM50; CM20
Request Format	audio set preset {"preset_name":"<value>" } \r\n
Return Format	audio set preset {"preset_name":"<value>" } \r\n

Sample	Send: audio set preset {"preset_name":"Custom1"} \r\n Return: audio set -preset {"preset_name":"Custom1"} \r\n
--------	---

Get Audio Mute Status

• Return Parameters

Name	Type	Value Range	Description
status	string	[on,off]	Mute Status on: Enabled off: Disabled

• Description

Audio command	Audio get mute
Supported Model	AP08; AVBridge; CM20; CM50; CS10; CS10-D
Request Format	audio get mute\r\n
Return Format	audio get mute {"status":"on"}\r\n
Sample	Send: audio get mute \r\n Return: audio get mute {"status":"on"} \r\n

Set Audio Mute Status

Set mute status for all channels of the device, which is not synchronized with the status obtained or set through the individual channel mute interface.

• Request Parameters

Name	Type	Value Range	Necessity	Description
status	string	[on,off]	Yes	Mute Status on: Enabled off: Disabled

• Description

Audio command	Audio set mute {"status":"<value>"}
Supported Model	AP08; AVBridge; CM20; CM50; CS10; CS10-D
Request Format	audio set mute {"status":"off"}\r\n
Return Format	audio set mute {"status":"off"}\r\n

Sample	Send: audio set mute {"status":"off"} \r\n Return: audio set mute {"status":"off"} \r\n
--------	--

Get Current Device Division Status

- Return Parameters

Name	Type	Value Range	Necessity	Description
room_info	room_info[]	-	Yes	Room information

room_info

Name	Type	Value Range	Necessity	Description
room_id	int[]	[1,3]	Yes	ID of the smallest divisible rooms contained in each divisible room

- Description

division-room	division-room get config
Supported Model	AP08
Request Format	division-room get config\r\n
Return Format	division-room get config {"room_info":{"room_id":[1]},{"room_id":[2]},{"room_id":[3]}}\r\n
Sample	Send: division-room get config\r\n Return: division-room get config {"room_info":{"room_id":[1]},{"room_id":[2]},{"room_id":[3]}}\r\n

Set the Current Division State of the Device

- Request Parameters

Name	Type	Value Range	Necessity	Description
room_info	room_info[]	-	Yes	Room information

room_info

Name	Type	Value Range	Necessity	Description
room_id	int[]	[1,3]	Yes	ID of the smallest divisible rooms contained in each divisible room

- **Description**

Division-room command	Division-room set config
Supported Model	AP08
Request Format	division-room set config {"room_info":{"room_id":[1,2]},{"room_id":[3]}}\r\n
Return Format	division-room set config {"room_info":{"room_id":[1,2]},{"room_id":[3]}}\r\n
Sample	Send: division-room set config {"room_info":{"room_id":[1,2]},{"room_id":[3]}}\r\n Return: division-room set config {"room_info":{"room_id":[1,2]},{"room_id":[3]}}\r\n

Get Hardware Information

- **Return Parameters**

Name	Type	Value Range	Description
hardware_list	body	[object]	Yes
type	body	string	Yes
index	body	[string]	No

- **Description**

System command	System get hardware
Supported Model	AP08
Request Format	system get hardware {"hal_list":{"type":<value>,"mode":<value>,"index":[<value>]}}\r\n
Return Format	system get hardware {"hal_list":{"type":<value>,"mode":<value>,"status":{"index":<value>,"status":<value>}}}\r\n
Sample	Send: system get hardware {"hal_list":{"type":"gpio","mode":"input","index":[1,2]}}\r\n Return: system get hardware {"hal_list":{"type":"gpio","mode":"input","status":{"index":1,"status":"high"},"index":2,"status":"low"}}}\r\n

Get Channel Volume

- **Request Parameters**

Name	Type	Value Range	Necessity	Description
------	------	-------------	-----------	-------------

arr_gain_info	string[]	-	Yes	Channel name array, and gain is a physical quantity. This API requires specifying a channel name; if no channel name is provided, it will return the gain value.
---------------	----------	---	-----	--

- Return Parameters

Name	Type	Value Range	Necessity	Description
arr_gain_info	arr_gain_info[]	-	Yes	-

arr_gain_info

Name	Type	Value Range	Description
name	string	-	Channel Name
value	int	[0-100]	Channel volume may have discrepancies. If you set it to 50, you might get 51.

- Description

Audio command	Audio get channel volume
Supported Model	AP08; AVBridge; CM50; CM20
Request Format	audio get channel volume {"arr_gain_info":["Dante-Input-1","Dante-Input-2"]}\r\n
Return Format	audio get channel volume {"arr_gain_info":[{"name":"Dante-Input-1","value":51}, {"name":"Dante-Input-2","value":60}]} \r\n CM20: audio get channel volume {"arr_gain_info":["Microphone-Input-1"]}\r\n
Sample	Send: audio get channel volume\r\n Return: audio get channel volume {"arr_gain_info":[{"name":"Dante-Input-1","value":51}, {"name":"Dante-Input-2","value":60}]} \r\n CM20: audio get channel volume {"arr_gain_info":["Microphone-Input-1"]}\r\n

Set Channel Volume

- Request Parameters

Name	Type	Value Range	Necessity	Description
arr_gain_info	arr_gain_info[]	-	Yes	-

arr_gain_info

Name	Type	Value Range	Description
name	string	-	Channel Name
value	int	[0-100]	Channel volume may have discrepancies. If you set it to 50, you might get 51.

- **Description**

Audio command	Audio set channel volume
Supported Model	AP08; AVBridge; CM50; CM20
Request Format	audio set channel volume {"arr_gain_info":[{"name":"Dante-Input-1","value":50}, {"name":"Dante-Input-2","value":60}]} \r\n
Return Format	audio set channel volume {"arr_gain_info":[{"name":"Dante-Input-1","value":50}, {"name":"Dante-Input-2","value":60}]} \r\n
Sample	Send: audio set channel volume {"arr_gain_info":[{"name":"Dante-Input-1","value":50}, {"name":"Dante-Input-2","value":60}]} \r\n Return: audio set channel volume {"arr_gain_info":[{"name":"Dante-Input-1","value":50}, {"name":"Dante-Input-2","value":60}]} \r\n

Get Channel Mute Status

- **Request Parameters**

Name	Type	Required	Constraint	Chinese Name	Description
channel_mute	[object]	true	none	Channel Mute Information List	none

- **Return Parameters**

Name	Type	Required	Constraint	Chinese Name	Description
channel_mute	[object]	true	none	Channel Mute Information List	none
name	string	true	none	Channel Name	Channel name displayed on the Designer UI
status	string	true	[off, on]	mute status	none

- **Description**

Audio command	Audio get channel mute {"channel_mute":{"<value>"}}
Supported Model	AP08; AVBridge; CM50; CM20
Request Format	audio get channel mute {"channel_mute":{"<value>"}} \r\n
Return Format	audio get channel mute {"channel_mute":[{"name":"<value>","status":<value>}]} \r\n
Sample	Send: audio get channel mute {"channel_mute":["Pickup-Area-1","Local-Amplification-Out"]} \r\n Return: audio get channel mute {"channel_mute":[{"name":"Pickup-Area-1","status":"on"}, {"name":"Local-Amplification-Out","status":"off"}]} \r\n

Set Channel Mute Status

• Request Parameters

Name	Type	Required	Constraint	Chinese Name	Description
channel_mute	[object]	true	none	Channel Mute Information List	none
name	string	true	none	Channel Name	Channel name displayed on the Designer UI
status	string	true	[off, on]	mute status	none

• Description

Audio command	Audio set channel mute {"channel_mute":[{"name":"<value>","status":<value>}]}
Supported Model	AP08; AVBridge; CM50; CM20
Request Format	audio set channel mute {"channel_mute":[{"name":"<value>","value":<value>}]} \r\n
Return Format	audio set channel mute {} \r\n
Sample	Send: audio set channel mute {"channel_mute":[{"name":"Pickup-Area-1","status":"on"}, {"name":"Amplifier-Area-1","value":"off"}]} \r\n Return: audio set channel mute {"channel_mute":[{"name":"Pickup-Area-1","status":"on"}, {"name":"Amplifier-Area-1","value":"off"}]} \r\n

Get Current Gain

• Request Parameters

Name	Type	Required	Constraint	Chinese Name	Description
arr_gain_info	string[]	true	none	Channel Gain Information List	none

• Return Parameters

Name	Type	Required	Constraint	Chinese Name	Description
arr_gain_info	[object]	true	none	Channel Gain Information List	none
name	string	true	none	Channel Name	Channel name displayed on the Designer UI
value	integer	true	[-60, 20]	Gain Value	none

• Description

Audio command	Audio get channel gain {"arr_gain_info":["<value>"]}
Supported Model	AP08; AVBridge; CM50; CM20
Request Format	audio get channel gain {"arr_gain_info":["<value>"]} \r\n
Return Format	audio get channel gain {"arr_gain_info":{"name":"<value>","value":<value>}} \r\n
Sample	Send: audio get channel gain {"arr_gain_info":["USB-Input-1","USB-Input-2"]} \r\n Return: audio get channel gain {"arr_gain_info":{"name":"USB-Input-1","value":10},{"name":"USB-Input-2","value":10}} \r\n Note: If the parameter is not filled in, the gain value of all channels will be returned.

Set Channel Gain

• Request Parameters

Name	Type	Required	Constraint	Chinese Name	Description
arr_gain_info	[object]	true	none	Channel Gain Information List	none
name	string	true	none	Channel Name	Channel name displayed on the Designer UI
value	integer	true	[-60, 20]	Gain Value	none

- Description

Audio command	Audio set channel gain {"arr_gain_info":[{"name":"<value>","value":<value>}]}
Supported Model	AP08; AVBridge; CM50; CM20
Request Format	audio set channel gain {"arr_gain_info":[{"name":"<value>","value":<value>}]} \r\n
Return Format	audio set channel gain {"arr_gain_info":[{"name":"<value>","value":<value>}]} \r\n
Sample	Send: audio set channel gain {"arr_gain_info":[{"name":"USB-Input-1","value":10}, {"name":"USB-Input-2","value":10}]} \r\n Return: audio set channel gain {"arr_gain_info":[{"name":"USB-Input-1","value":10}, {"name":"USB-Input-2","value":10}]} \r\n