MIRI

V410 RESTful API Specification

Preview Edition for Integration Testing

Version	1.0
Target Hardware	MIRI V410 Video Encoder / Decoder
Interface	HTTP RESTful API with JSON payload
Encoding	UTF-8

Purpose of this preview

This edition is being shared with integrators and end users to validate the API in real-world workflows. The technical API content is presented as supplied for testing. Please report unclear behavior, unexpected responses, missing fields, or integration issues to Miri Technologies.

Feedback is especially valuable when it includes the endpoint, request payload, device firmware version, expected result, and actual result.

PREVIEW / FEEDBACK EDITION
This document is intended for integration testing and feedback before final public release.

Contents

Overview & Authentication
- Authentication Flow
- 1. User Login / Obtain AuthToken
- 2. Request Header Specification
- 3. Session Expiration & Re-authentication

1. Video & Audio Encoding Module
- 1.1 Select Video Source
- 1.2 Get Video Source Status & Information
- 1.3 WebRTC Encoding Preview
- 1.4 Get Audio Volume Levels (dB)
- 1.5 Get Video Encoding Status
- 1.6 Get Audio & Video Encoding Parameters
- 1.7 Set Video Encoding Parameters
- 1.8 Get Supported Video Encoding Resolutions
- 1.9 Set Audio Encoding Parameters
- 1.10 Upload OSD Image
- 1.11 Delete OSD Image
- 1.12 Get OSD Image List
- 1.13 Add OSD Overlay
- 1.14 Delete OSD Overlay
- 1.15 Modify OSD Overlay
- 1.16 Set Master OSD Overlay Switch
- 1.17 Toggle Individual OSD Overlay
- 1.18 Get OSD Overlay List
- 1.19 Upload No-Signal Background Image
- 1.20 Delete No-Signal Background Image
- 1.21 Get No-Signal Background Image List
- 1.22 Set Active No-Signal Background Image
- 1.23 Set Video Cropping
- 1.24 Get Video Cropping Settings
- 1.25 Set Rotation, Mirroring, and Flipping
- 1.26 Get Rotation, Mirroring, and Flipping

2. Streaming Service Module
- 2.1 Add Streaming Service
  - 2.1.1 SRT Service
  - 2.1.2 NDI®|HX2 Service
  - 2.1.3 NDI®|HX3 Service
  - 2.1.4 RTSP Server Service
  - 2.1.5 HLS Server Service
  - 2.1.6 TS over UDP Service
  - 2.1.7 RTMP Server / Push Service
  - 2.1.8 NDI High Bandwidth (NDI HB) Service
  - 2.1.9 OMT Service
- 2.2 Delete Streaming Service
- 2.3 Modify Streaming Service
- 2.4 Start / Stop Streaming Service
- 2.5 Get Streaming Service List

3. Decoding Module
- 3.1 Add Decode Source
- 3.2 Delete Decode Source
- 3.3 Modify Decode Source

3.4 Get Decode Source List
3.5 Select Decode Output Source
3.6 Remove / Close Decode Output Source
3.7 Get Real-Time Decoding Status
3.8 Get Output Interface Configuration
3.9 Set Output Interface Configuration
3.10 Get Supported Output Interface Options & Ranges
3.11 Get Output Audio Volume Levels (dB)
3.12 WebRTC Decoding Preview
3.13 Get NDI Network Discovery List
3.14 Get OMT Network Discovery List

4. Recording Module

4.1 Get Recording Configuration
4.2 Get External Storage Device Information
4.3 Set Recording Configuration
4.4 Get Recording Status
4.5 Start Recording
4.6 Stop Recording
4.7 Get Recorded File List from Device
4.8 Eject Storage Device

5. System Module

5.1 Get Advanced System Settings
5.2 Configure ZeroTier VPN
5.3 Configure SNMP Service
5.4 Configure SSDP Service
5.5 Set USB Type-C Operating Mode
5.6 Get Analog Audio Output Routing
5.7 Set Analog Audio Output Routing
5.8 Configure NDI Discovery Server
5.9 Get NDI Discovery Server Configuration
5.10 Reboot Device
5.11 Restore Factory Defaults
5.12 Get LED Status & Configuration
5.13 Set LED Status & Configuration
5.14 Get Real-Time System Performance Status
5.15 Get System Hardware & Firmware Information

6. Network Module

6.1 Get Network Link Configuration
6.2 Configure Network Connection
6.3 Get USB Network Interface Information
6.4 Set USB Network IP Address
6.5 Get Wi-Fi Status
6.6 Enable / Disable Wi-Fi
6.7 Get Scanned Wi-Fi Networks List
6.8 Connect to Wi-Fi Network
6.9 Connect to Hidden Wi-Fi Network
6.10 Disconnect / Forget Wi-Fi Network

7. Scheduled Tasks Module

7.1 Get Scheduled Tasks
7.2 Create Scheduled Task
7.3 Update Scheduled Task
7.4 Enable / Disable Scheduled Task
7.5 Delete Scheduled Task

8. User Management Module

8.1 User Login
8.2 Toggle Two-Factor Authentication (2FA)
8.3 Activate Two-Factor Authentication (2FA)
8.4 Get User Status
Appendix: Error Codes & Status Reference

Overview & Authentication

Authentication Flow

Before executing any API commands, callers must complete the authentication handshake to obtain an AuthToken. All subsequent requests require this AuthToken to be passed in the HTTP request headers.

Token Expiration Policy:

- The AuthToken automatically expires after **30 minutes of inactivity** or immediately upon **device reboot**.
- When expired, requests return an authentication failure, requiring the client to re-login and obtain a fresh token.

1. User Login / Obtain AuthToken

Authenticates administrative credentials and returns an active session AuthToken.

Endpoint: `/system/user/login.json`
Method: `POST`

Request Payload

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

Response Payload

```
{
 "result": "ok",
 "data": {
  "username": "admin",
  "alias": "admin",
  "AuthToken": "71e92e318c23f2c30ca8feb4c207a5d6"
 }
}
```

2. Request Header Specification

Every authenticated API request must include the app header carrying the username and AuthToken formatted as a serialized JSON string:

```
app: {"username":"admin","AuthToken":"71e92e318c23f2c30ca8feb4c207a5d6"}
```

Ensure this header is present in all subsequent HTTP calls to ensure successful access authorization.

3. Session Expiration & Re-authentication

If the AuthToken is expired, absent, or invalid, the server responds with:

```
{
 "result": "auth-failed",
 "error": "Not exists token infomation."
}
```

1. Video & Audio Encoding Module

The Encoding Module governs video capture source selection, encoding configurations (H.264, H.265), real-time WebRTC preview generation, audio channel routing, OSD text/clock/image overlays, video cropping, and rotation/mirror controls.

1.1 Select Video Source

Assigns an input video source to the specified encoder channel or disables encoding.

Endpoint: `/codec/encode/select_source.json`
Method: `POST`

Request Payload

```
{
 "encode_id": 2,
 "source": "",
 "url": "192.168.0.91:5961",
 "name": "DESKTOP-85GQQJV (VLC)"
}
```

Parameter Description

Parameter	Type	Required	Description
encode_id	Integer	Yes	Target encoder channel ID: 1 or 2.
source	String	Yes	Video input source type: ndi, omt, usb, hdmi, sdi. An empty string "" disables encoding.
url	String	No	NDI source connection address (host:port). Required when source="ndi". (See Section 3.1).
name	String	No	Source stream identifier. Required when source="ndi" or source="omt".

Response Payload

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

1.2 Get Video Source Status & Information

Queries hardware signal status, input resolution, refresh rate, and audio properties for a specific encoder channel.

Endpoint: `/codec/encode/get_source_status.json`
Method: `POST`

Request Payload

```
{
 "encode_id": 1
}
```

Parameter Description

Parameter	Type	Required	Description
encode_id	Integer	Yes	Encoder channel ID: 1 or 2.

Response Payload

```
{
 "data": {
  "name": "A1-557E (GAME_NDI HX2)",
  "fps": 59.94,
  "source": "ndi",
  "height": 1080,
  "url": "192.168.0.177:5961",
  "width": 1920,
  "signal": 1,
  "sample_rate": 48000,
  "ip": "192.168.0.177",
  "interlaced": false,
  "resolution": "1920x1080P 59.94Hz"
 },
 "result": "ok"
}
```

Field Description

Field	Type	Description
source	String	Active input source: hdmi, sdi, usb, ndi, omt.
sample_rate	Integer	Source audio sampling frequency in Hz (e.g., 48000).
width	Integer	Input video frame width in pixels.
height	Integer	Input video frame height in pixels.
fps	Float	Input video frame rate.
signal	Integer	Signal lock status: 0: No Signal, 1: Signal Locked, 2: Connecting, 3: Reconnecting, 4: Unsupported Resolution.
interlaced	Boolean	Scan format flag: true for interlaced (e.g., 1080i), false for progressive (1080p).
resolution	String	Formatted signal string (e.g., "1920x1080P 59.94Hz").
name	String	Source stream name (only present when source is ndi or omt).
ip	String	Source IP address (NDI source).
url	String	Full connection URL / address string.

1.3 WebRTC Encoding Preview

Establishes a low-latency WebRTC live preview stream or single frame snapshot for real-time monitoring on a web interface.

Endpoint: `/codec/webrtc/add_webrtc.json`
Method: `POST` / `GET`

Request Payload

```
{
 "mode": 0,          // 0: Video stream mode, 1: Snapshot image mode
 "ip": "192.168.1.1",   // Client IP address requesting preview
 "encode_id": 1
}
```

Parameter Description

Parameter	Type	Required	Description
mode	Integer	Yes	Stream mode: 0: Real-time video preview, 1: Snapshot image mode.
ip	String	Yes	IP address of the client browser / consumer.
encode_id	Integer	Yes	Encoder channel ID: 1 or 2.

Response Payload

```
{
 "result": "ok",
 "data": {
   "port": 51545,
   "sdp": "v=0\r\no=- 1495799811084970 2 IN IP4 192.168.0.176\r\ns=-\r\nt=0 0\r\na=group:BUNDLE video\r\na=ice-lite
\r\nm=video 9 UDP/TLS/RTP/SAVPF 126\r\nc=IN IP4 0.0.0.0\r\na=ice-ufrag:uoxv\r\na=ice-
pwd:93urglck0zhkz1q4a87clihx\r\na=ice-options:trickle\r\na=fingerprint:sha-256
21:39:6D:25:62:F6:14:A5:28:95:39:DF:7D:F9:DC:D3:13:FA:93:B9:57:54:19:60:62:F9:CF:51:D6:69:DC:D5\r\na=setup:actpass\r\na=
mid:video\r\na=sendonly\r\na=rtcp-mux\r\na=rtpmap:126 H264/90000\r\na=fmtp:126 level-asymmetry-
allowed=1;packetization-mode=1;profile-level-id=42e01f\r\na=candidate:1 1 udp 12345678 192.168.0.176 51545 typ
host\r\na=end-of-candidate\r\n"
 }
}
```

1.4 Get Audio Volume Levels (dB)

Returns real-time audio volume levels (in decibels) for the active encoder channel.

Endpoint: `/codec/encode/get_encode_real_db.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1
}
```

Response Payload

```
{
 "result": "ok",
 "data": [
   -50.905269, // Left channel volume in dB
   -50.891619  // Right channel volume in dB
 ]
}
```

1.5 Get Video Encoding Status

Queries current runtime video and audio encoding telemetry.

Endpoint: `/codec/encode/get_encode_status.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1
}
```

Response Payload

```
{
 "data": {
  "fps": 60,
  "bitrate": 4307,
  "channels": 2,
  "source": "hdmi",
  "sample_rate": 48000,
  "codec": "H.264",
  "height": 1080,
  "width": 1920
 },
 "result": "ok"
}
```

Field Description

Field	Type	Description
width	Integer	Encoding output frame width (pixels).
height	Integer	Encoding output frame height (pixels).
fps	Integer	Encoding output frame rate (fps).
source	String	Source input interface name (hdmi, sdi, etc.).
sample_rate	Integer	Audio sampling frequency in Hz.
channels	Integer	Active encoded audio channels count.
codec	String	Video codec standard: H.264, H.265, SHQ2.
bitrate	Integer	Real-time encoding bitrate in Kbps.

1.6 Get Audio & Video Encoding Parameters

Retrieves the complete configuration profiles for video encoding and multi-channel audio processing.

Endpoint: `/codec/encode/get_encode_info.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1
}
```

Response Payload

```
{
 "data": {
  "video_setting": {
   "codec": "H264",
   "maxBitrate": 4096,
```

```
    "fps": 60,
    "width": 1920,
    "height": 1080,
    "enable_omt": 0,
    "enable_ndi_hx3": 0,
    "profile": "High",
    "rc": "CBR",
    "gop": 60,
    "bitrate": 4096,
    "same_vin": false
   },
   "audio_setting": {
    "bitrate": 128000,
    "channels": 2,
    "sample_rate": 48000,
    "source_list": [
     {
      "channels_map": [1, 2, 3, 4, 5, 6, 7, 8],
      "enable": true,
      "channels": 8,
      "source": "Encoder Source",
      "gain": 0
     },
     {
      "channels_map": [1, 2, 0, 0, 0, 0, 0, 0],
      "enable": false,
      "channels": 2,
      "source": "Line In",
      "gain": 0
     }
    ],
    "use_lfe": false
   }
  },
  "result": "ok"
}
```


Parameter Breakdown

Parameter	Type	Description
video_setting.codec	String	Video compression standard: `H265` or `H264`.
video_setting.width	Integer	Frame width in pixels.
video_setting.height	Integer	Frame height in pixels.
video_setting.fps	Integer	Target frame rate (fps).
video_setting.profile	String	Codec profile: `BaseLine`, `Main`, `High` (H.264 profile is restricted to `Main` in certain configs).
video_setting.gop	Integer	Group of Pictures (GOP) / Keyframe interval (in frames).
video_setting.same_vin	Boolean	Video passthrough: `true` to match input video dimensions/fps automatically.
video_setting.rc	String	Rate control algorithm: `CBR` (Constant Bitrate) or `VBR` (Variable Bitrate).
video_setting.bitrate	Integer	Target encoding bitrate in Kbps. Allowed range: `[1024, 40960]`.

video_setting.maxBitrate	Integer	Peak bitrate limit under VBR mode in Kbps. Allowed range: [1024, 40960].
video_setting.enable_ndi_hx3	Integer	NDI®|HX3 mode flag: 1: Enabled, 2: Disabled. (Note: When enabled, bitrate and GOP cannot be modified).
video_setting.enable_omt	Integer	NDI HB / OMT mode flag: 1: Enabled, 2: Disabled. (Note: When enabled, only resolution can be modified).
audio_setting.bitrate	Integer	Audio bitrate in bps: 32000, 48000, 64000, 96000, 128000, 192000, 256000.
audio_setting.channels	Integer	Encoded audio channel count: [1, 8].
audio_setting.sample_rate	Integer	Audio sample rate: 32000, 44100, or 48000 Hz.
audio_setting.use_lfe	Boolean	Low-Frequency Effects (LFE / subwoofer) enable flag.
audio_setting.source_list	Array	Available audio source devices list.
source_list[i].source	String	Audio input source name (e.g., "Encoder Source" for embedded audio, "Line In" for 3.5mm analog audio).
source_list[i].enable	Boolean	Source activation flag.
source_list[i].channels	Integer	Total usable channels available on this source.
source_list[i].gain	Integer	Volume gain adjustment in dB: [-40, 20].
source_list[i].channels_map	Array	8-channel routing matrix. Array index 0..7 denotes the output channel, and the integer value denotes the source channel mapped to it (e.g., channels_map[0]=2 routes source channel 2 into output channel 1).

1.7 Set Video Encoding Parameters

Configures video encoding parameters including resolution, frame rate, rate control, and GOP.

Endpoint: `/codec/encode/video_setting.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "same_vin": false,
 "width": 1920,      // Resolution must match one of the presets from Section 1.8
 "height": 1080,
 "fps": 30,
 "rc": "CBR",
 "profile": "High",
 "bitrate": 2048,    // Unit: kbps
 "maxBitrate": 10000,  // Unit: kbps (effective when rc="VBR")
 "gop": 60,
```

```
 "codec": "H264"
}
```

Response Payload

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

1.8 Get Supported Video Encoding Resolutions

Retrieves the list of valid video encoding resolution presets supported by the encoder hardware.

Endpoint: `/codec/encode/get_config_list.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1
}
```

Response Payload

```
{
 "data": {
  "resolution": [
   { "name": "encode.same_vin" },
   { "fps": 60, "name": "3840x2160P 60Hz", "width": 3840, "height": 2160 },
   { "fps": 50, "name": "3840x2160P 50Hz", "width": 3840, "height": 2160 },
   { "fps": 30, "name": "3840x2160P 30Hz", "width": 3840, "height": 2160 },
   { "fps": 60, "name": "2560x1440P 60Hz", "width": 2560, "height": 1440 },
   { "fps": 50, "name": "2560x1440P 50Hz", "width": 2560, "height": 1440 },
   { "fps": 30, "name": "2560x1440P 30Hz", "width": 2560, "height": 1440 },
   { "fps": 60, "name": "1920x1080P 60Hz", "width": 1920, "height": 1080 },
   { "fps": 50, "name": "1920x1080P 50Hz", "width": 1920, "height": 1080 },
   { "fps": 30, "name": "1920x1080P 30Hz", "width": 1920, "height": 1080 },
   { "fps": 25, "name": "1920x1080P 25Hz", "width": 1920, "height": 1080 },
   { "fps": 24, "name": "1920x1080P 24Hz", "width": 1920, "height": 1080 },
   { "fps": 60, "name": "1280x720P 60Hz",  "width": 1280, "height": 720 },
   { "fps": 50, "name": "1280x720P 50Hz",  "width": 1280, "height": 720 },
   { "fps": 30, "name": "1280x720P 30Hz",  "width": 1280, "height": 720 }
  ]
 },
 "result": "ok"
}
```

Note: The resolution specified in Section 1.7 MUST be chosen strictly from this supported preset list.

1.9 Set Audio Encoding Parameters

Configures audio sample rate, bitrate, channel count, and channel mapping for the encoder.

Endpoint: `/codec/encode/audio_setting.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "bitrate": 128000,
 "channels": 4,
 "sample_rate": 48000,
 "use_lfe": false,
 "source_list": [
  {
   "enable": true,
   "gain": 0,
   "channels_map": [1, 2, 3, 4, 5, 6, 7, 8]
```

```
  },
  {
   "enable": false,
   "gain": 0,
   "channels_map": [1, 2, 0, 0, 0, 0, 0, 0]
  }
 ]
}
```

Response Payload

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

1.10 Upload OSD Image

Uploads a custom image file (PNG/JPEG) to device storage for On-Screen Display overlay.

Endpoint: `/codec/img/add_image.json`
Method: `POST` (Multipart / Form-data)

Response Payload

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

1.11 Delete OSD Image

Removes an uploaded OSD image from the device storage.

Endpoint: `/codec/img/del_image.json`
Method: `POST` / `GET`

Request Payload

```
{
 "id": 1 // Image ID to delete
}
```

Response Payload

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

> **Warning:** Deleting an image currently bound to an active OSD layer will automatically remove that OSD overlay layer.

1.12 Get OSD Image List

Lists all uploaded OSD images stored on the device.

Endpoint: `/codec/img/get_images.json`
Method: `POST` / `GET`

Response Payload

```
{
 "data": [
  {
   "id": 1,
   "img_type": "jpeg",
   "width": 132,
   "height": 132,
   "file_size": 5011,
   "path": "/images/osd/1.jpeg"
  },
```

```
  {
    "id": 2,
    "img_type": "png",
    "width": 100,
    "height": 100,
    "file_size": 13985,
    "path": "/images/osd/2.png"
  },
  {
    "id": 3,
    "img_type": "png",
    "width": 1920,
    "height": 1080,
    "file_size": 310784,
    "path": "/images/osd/3.png"
  }
 ],
 "result": "ok"
}
```


1.13 Add OSD Overlay

Creates a new On-Screen Display (OSD) overlay on an encoder channel. Supports timestamps, custom text, and graphic watermarks.

Endpoint: `/codec/encode/add_surface.json`

Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "class": 1,         // 1: Timestamp / Clock, 2: Text, 3: Image
 "position": 1,      // 0: Custom, 1: Top-Left, 2: Top-Center, 3: Top-Right, 4: Middle-Left, 5: Center, 6: Middle-Right, 7: Bottom-Left, 8: Bottom-Center, 9: Bottom-Right
 "x_offset": 0,      // X coordinate offset in pixels (valid when position=0)
 "y_offset": 0,      // Y coordinate offset in pixels (valid when position=0)
 "format": 0,        // Date/Time format (valid when class=1)
 "text": "",         // Custom text string (valid when class=2)
 "font_size": 20,    // Font size: [12, 72]
 "font_color": 16711680, // Decimal RGB color (e.g., 0xFF0000 = 16711680 for Red)
 "image_id": 1       // Image ID from Section 1.12 (valid when class=3)
}
```

Parameter Description

Parameter	Type	Required	Description
encode_id	Integer	Yes	Target channel ID: 1: Game / Channel 1, 2: Camera / Channel 2, 3: Mixed output.
class	Integer	Yes	OSD Element Type: 1: Timestamp / Clock, 2: Text banner, 3: Graphic image.
position	Integer	Yes	Anchor position: 0: Custom (uses offsets), 1: Top-Left, 2: Top-Center, 3: Top-Right, 4: Middle-Left, 5: Center, 6: Middle-Right, 7: Bottom-Left, 8: Bottom-Center, 9: Bottom-Right.
x_offset	Integer	No	Horizontal offset in pixels.

y_offset	Integer	No	Vertical offset in pixels.
format	Integer	No	Clock format (when class=1): 0: YYYY-MM-DD hh:mm:ss 1: MM/DD/YYYY hh:mm:ss 2: DD/MM/YYYY hh:mm:ss 3: YYYY-MM-DD 4: MM/DD/YYYY 5: DD/MM/YYYY 6: hh:mm:ss 7: hh:mm
text	String	No	Text string content (when class=2).
font_size	Integer	No	Font size in points: [12, 72].
font_color	Integer	No	Color represented as 24-bit decimal RGB integer (e.g., 0xFF0000 = 16711680).
image_id	Integer	No	Image ID from Section 1.12 (when class=3).

Response Payload

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

1.14 Delete OSD Overlay

Deletes an active OSD overlay item by ID.

Endpoint: `/codec/encode/del_surface.json`
Method: `POST` / `GET`

Request Payload

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

Response Payload

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

1.15 Modify OSD Overlay

Updates an existing OSD overlay layer's properties and transparency.

Endpoint: `/codec/encode/set_surface.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "id": 1,
```

```
 "class": 1,
 "position": 1,
 "x_offset": 0,
 "y_offset": 0,
 "format": 0,
 "text": "",
 "font_size": 20,
 "font_color": 16711680,
 "image_id": 1,
 "alpha": 255         // Transparency level: [0, 255] (255 = completely opaque)
}
```

Response Payload

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

1.16 Set Master OSD Overlay Switch

Enables or disables all OSD overlays globally for the given encoder channel.

Endpoint: `/codec/encode/set_enable_surface.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "enable": false
}
```

Response Payload

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

1.17 Toggle Individual OSD Overlay

Enables or disables a specific OSD overlay layer without altering other layers.

Endpoint: `/codec/encode/enable_surface.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "id": 1,
 "enable": false
}
```

Response Payload

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

1.18 Get OSD Overlay List

Queries all configured OSD overlay items and the master switch state.

Endpoint: `/codec/encode/get_surface_all.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1
```

```
}
```

Response Payload

```
{
 "data": {
  "enable": true,
  "surfaces": [
   {
    "id": 1,
    "class": 1,
    "position": 1,
    "x_offest": 0,
    "y_offest": 0,
    "format": 0,
    "font_size": 20,
    "font_color": 16711680,
    "image_id": 1,
    "alpha": 255,
    "enable": true,
    "text": "abcd"
   }
  ]
 },
 "result": "ok"
}
```

1.19 Upload No-Signal Background Image

Uploads a custom fallback background image to display when video input signal is lost.

Endpoint: `/codec/img/add_nosignal_file.json`
Method: `POST` / `GET` (Multipart Form-Data)

Request Payload

```
{
 "upload": "binary_image_file",
 "source": "hdmi" // Target source interface: "hdmi" or "sdi"
}
```

Response Payload

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

1.20 Delete No-Signal Background Image

Deletes an uploaded no-signal fallback image.

Endpoint: `/codec/img/del_nosignal_file.json`
Method: `POST` / `GET`

Request Payload

```
{
 "source": "hdmi", // Target interface: "hdmi" or "sdi"
 "id": 3          // Image ID to delete (IDs 1 and 2 are system defaults and protected)
}
```

Response Payload

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

Note: System default images (id: 1 and id: 2) are protected and cannot be deleted.

1.21 Get No-Signal Background Image List

Retrieves the list of available no-signal placeholder images for HDMI and SDI interfaces.

Endpoint: `/codec/img/get_nosignal_files.json`
Method: `POST` / `GET`

Response Payload


```
{
 "data": {
  "hdmi": [
   {
    "id": 1,
    "width": 1920,
    "height": 1080,
    "img_type": "png",
    "path": "/images/nosignal/1.png",
    "no_delete": true,
    "is_used": false
   },
   {
    "id": 2,
    "width": 1920,
    "height": 1080,
    "img_type": "png",
    "path": "/images/nosignal/2.png",
    "no_delete": true,
    "is_used": true
   }
  ],
  "sdi": [
   {
    "id": 1,
    "width": 1920,
    "height": 1080,
    "img_type": "png",
    "path": "/images/nosignal/1.png",
    "no_delete": true,
    "is_used": false
   },
   {
    "id": 2,
    "width": 1920,
    "height": 1080,
    "img_type": "png",
    "path": "/images/nosignal/2.png",
    "no_delete": true,
    "is_used": true
   }
  ]
 },
 "result": "ok"
}
```


1.22 Set Active No-Signal Background Image

Selects the active no-signal fallback graphic for HDMI or SDI input.

Endpoint: `/codec/img/set_nosignal_file.json`
Method: `POST` / `GET`

Request Payload

```
{
 "source": "sdi", // "hdmi" or "sdi"
 "id": 1
}
```

Response Payload

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

1.23 Set Video Cropping

Configures rectangular region-of-interest cropping on the video input.

Endpoint: `/codec/encode/set_crop.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "crop_x": 25,
 "crop_y": 25,
 "crop_w": 50,
 "crop_h": 50
}
```

Parameter Description

Parameter	Type	Required	Description
encode_id	Integer	Yes	Target encoder channel ID.
crop_x	Integer	Yes	Horizontal start percentage [0, 100].
crop_y	Integer	Yes	Vertical start percentage [0, 100].
crop_w	Integer	Yes	Width crop percentage. Condition: crop_x + crop_w <= 100.
crop_h	Integer	Yes	Height crop percentage. Condition: crop_y + crop_h <= 100.

Response Payload

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

1.24 Get Video Cropping Settings

Queries the current video cropping parameters and status.

Endpoint: `/codec/encode/get_crop.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1
}
```

Response Payload

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

```
  "crop_x": 25,
  "crop_y": 25,
  "crop_w": 50,
  "crop_h": 50,
  "crop_en": false
 }
}
```

1.25 Set Rotation, Mirroring, and Flipping

Configures video orientation transforms (rotation, horizontal mirror, vertical flip).

Endpoint: `/codec/encode/set_rmf.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "rotation": 1,     // 0: 0°, 1: 90°, 2: 180°, 3: 270°
 "mirror_en": false, // Horizontal mirror
 "flip_en": false    // Vertical flip
}
```

Parameter Description

Parameter	Type	Required	Description
encode_id	Integer	Yes	Target encoder channel ID.
rotation	Integer	Yes	Clockwise rotation angle: 0: 0°, 1: 90°, 2: 180°, 3: 270°.
mirror_en	Boolean	Yes	Horizontal mirror reflection flag (true/false).
flip_en	Boolean	Yes	Vertical inversion / flip flag (true/false).

Response Payload

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

1.26 Get Rotation, Mirroring, and Flipping

Queries active video orientation transforms.

Endpoint: `/codec/encode/get_rmf.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1
}
```

Response Payload

```
{
 "result": "ok",
 "data": {
  "rotation": 0,
  "mirror_en": false,
  "flip_en": false
 }
```

```
}
```

2. Streaming Service Module

The Streaming Service Module manages live egress streaming pipelines, supporting SRT, NDI®|HX2, NDI®|HX3, RTSP, HLS, TS over UDP, RTMP, Full NDI (HB), and OMT protocols.

2.1 Add Streaming Service

Instantiates a new output streaming service bound to an encoder channel. The payload varies by protocol.

Endpoint: `/codec/stream/add.json`
Method: `POST` / `GET`

2.1.1 SRT Service

```
{
 "type": "srt",
 "name": "SRT",
 "mode": "Caller",                 // "Caller", "Listener", or "Rendezvous"
 "port": 1025,                     // Port number
 "latency": 125,                   // Buffer latency in ms
 "encryption": "AES-128",          // "AES-128", "AES-192", "AES-256", or empty
 "streamid": "123",                // Stream ID
 "multiple": 1,                    // Null packet stuffing multiplier: 1, 1.2, 1.5 (1 = disabled)
 "host": "192.168.0.11",           // Destination IP (required for Caller/Rendezvous mode)
 "passphrase": "saldkfjlsadkjfls",    // Pre-shared key (min 10 characters when encryption enabled)
 "encode_id": 1
}
```

2.1.2 NDI®|HX2 Service

```
{
 "type": "ndihx2",
 "name": "NDI HX2",
 "group_name": "",                 // NDI Group
 "channel_name": "NDI HX2",        // Stream name published over NDI
 "connection": "multicast",        // "tcp", "rudp", or "multicast"
 "netprefix": "239.255.0.0",       // Multicast subnet prefix (when connection="multicast")
 "netmask": "255.255.0.0",         // Multicast subnet mask (when connection="multicast")
 "ttl": 1,                         // Multicast Time-to-Live
 "encode_id": 1
}
```

2.1.3 NDI®|HX3 Service

```
{
 "type": "ndihx3",
 "name": "NDI HX3",
 "group_name": "",
 "channel_name": "NDI HX3",
 "connection": "multicast",
 "netprefix": "239.255.0.0",
 "netmask": "255.255.0.0",
 "ttl": 1,
 "encode_id": 1
}
```

2.1.4 RTSP Server Service

```
{
 "type": "rtsp",
 "name": "RTSP Server",
 "port": 554,                      // RTSP listening port
 "session": "ch01",                // Mount path session name (rtsp://<ip>:<port>/ch01)
 "user": "",                       // Optional authentication username
 "pass": "",                       // Optional authentication password
 "encode_id": 1
```

```
}
```

2.1.5 HLS Server Service

```
{
 "type": "hls",
 "name": "HLS Server",
 "location": "encode1",            // URL path mount name (http://<ip>/hls/encode1.m3u8)
 "max_files": 6,                   // Circular playlist window size: [5, 8]
 "duration": 10,                   // Target segment duration in seconds: [1, 15]
 "encode_id": 1
}
```

2.1.6 TS over UDP Service

```
{
 "type": "udp",
 "name": "TS over UDP",
 "host": "192.168.0.11",           // Destination unicast or multicast IP address
 "port": 1,                        // Destination port
 "multiple": 1,                    // Null packet insertion multiplier: 1, 1.2, 1.5 (1 = disabled)
 "alignment": -1,                  // TS packet alignment: -1 (Auto) or 7 (7 TS packets per UDP datagram)
 "encode_id": 1
}
```

2.1.7 RTMP Server / Push Service

```
{
 "type": "rtmp",
 "name": "RTMP Server",
 "location": "192.168.0.111",      // RTMP ingest URL or host
 "stream_key": "123",              // Live stream key
 "user": "",                       // Optional authentication username
 "pass": "",                       // Optional authentication password
 "encode_id": 1
}
```

2.1.8 NDI High Bandwidth (NDI HB) Service

```
{
 "type": "ndi",
 "name": "NDI HB",
 "channel_name": "NDI HB",
 "connection": "multicast",        // "tcp", "rudp", or "multicast"
 "netprefix": "239.255.0.0",
 "netmask": "255.255.0.0",
 "ttl": 1,
 "encode_id": 2
}
```

2.1.9 OMT Service

```
{
 "type": "omt",
 "name": "OMT",
 "channel_name": "OMT",
 "quality": 0,                     // Quality setting: 0: Auto, 1: Low, 50: Medium, 100: High
 "encode_id": 2
}
```

Response Payload

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

2.2 Delete Streaming Service

Deletes an existing stream output service by its unique UUID.

Endpoint: `/codec/stream/delete.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "index": "702acb69fc644c58865a440c3e6c31c2"
}
```

Response Payload

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

2.3 Modify Streaming Service

Modifies parameters of a previously configured streaming service.

Endpoint: `/codec/stream/modify_stream.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "index": "702acb69fc644c58865a440c3e6c31c2",
 // Include updated protocol parameters matching Section 2.1
}
```

Response Payload

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

2.4 Start / Stop Streaming Service

Controls the runtime execution state (enable/disable) of an active streaming service.

Endpoints

Start: `/codec/stream/start_stream.json`
Stop: `/codec/stream/stop_stream.json`
Method: `POST` / `GET`

Request Payload

```
{
 "encode_id": 1,
 "index": "702acb69fc644c58865a440c3e6c31c2"
}
```

Response Payload

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

2.5 Get Streaming Service List

Queries all configured streaming services and their real-time uptime status.

Endpoint: `/codec/stream/get.json`
Method: `POST` / `GET`

Request Payload


```
{
 "encode_id": 1
}
```


Response Payload


```
{
 "data": [
  {
   "id": "8bf62ee9b79a452b8f16e6e3f02ab8fe",
   "index": "8bf62ee9b79a452b8f16e6e3f02ab8fe",
   "type": "ndi",
   "name": "NDI HB",
   "channel_name": "NDI HB",
   "group_name": "",
   "connection": "multicast",
   "netprefix": "239.255.0.0",
   "netmask": "255.255.0.0",
   "ttl": 1,
   "width": 1920,
   "height": 1080,
   "fps": 30,
   "interlaced": 0,
   "enable": false,
   "up_time": 0,
   "status": []
  },
  {
   "id": "702acb69fc644c58865a440c3e6c31c2",
   "index": "702acb69fc644c58865a440c3e6c31c2",
   "type": "omt",
   "name": "OMT",
   "channel_name": "OMT",
   "quality": 0,
   "discovery": "",
   "width": 1920,
   "height": 1080,
   "fps": 60,
   "interlaced": 0,
   "enable": false,
   "up_time": 0,
   "status": []
  }
 ],
 "result": "ok"
}
```

3. Decoding Module

The Decoding Module governs ingress stream pulling and decoding to baseband video outputs (HDMI / SDI), video format conversions, audio routing, and network discovery of NDI/OMT sources.

3.1 Add Decode Source

Registers an incoming network stream source in the device's decode source pool.

Endpoint: `/codec/source/add.json`

Method: `POST` / `GET`

Source Protocol Examples

NDI Source:

```
{
 "type": "ndi",
 "host": "ENCODER57",
 "url": "192.168.0.57%3A5961", // URL-encoded host:port
 "name": "ENCODER57 (ndi)",
 "group": "public",
 "channel": "ndi"
}
```

OMT Source:

```
{
 "type": "omt",
 "name": "PC-202408271729 (vMix - Output 1)",
 "url": ""
}
```

RTSP Source:

```
{
 "type": "rtsp",
 "name": "RTSP Stream",
 "url": "rtsp://192.168.0.57:554/ch01",
 "transport_protocol": 7,  // 1: UDP, 2: UDP Multicast, 4: TCP, 7: UDP+TCP+UDP Multicast
 "network_caching": 200,   // Buffer latency in ms
 "rtsp_contimeout": 15,    // Connection timeout in seconds
 "rtsp_user": "admin",
 "rtsp_pwd": "password"
}
```

SRT Source:

```
{
 "type": "srt",
 "name": "SRT Stream",
 "url": "192.168.0.57",     // Effective in caller/rendezvous mode
 "port": "1234",
 "srt_mode": "caller",      // "caller", "listener", or "rendezvous"
 "srt_streamid": "abcd",    // Effective in listener mode
 "srt_passphrase": "",
 "srt_keylen": 0,
 "srt_latency": 125,
 "ts_latency": 200,
 "network_caching": 200
}
```

RTMP Source:

```
{
 "type": "rtmp",
 "name": "RTMP Pull",
 "url": "rtmp://192.168.0.57/live",
```

```
 "rtmp_stream_key": "stream1",
 "rtmp_user": "user",
 "rtmp_pwd": "pwd",
 "contimeout": 15,         // Timeout in seconds (default 10)
 "network_caching": 200
}
```

HLS Source:

```
{
 "type": "hls",
 "name": "HLS Stream",
 "url": "http://192.168.0.57/live/stream.m3u8",
 "islive": true,
 "timeout": 15,
 "user_id": "user",
 "user_pwd": "pwd",
 "network_caching": 200
}
```

TS over UDP Source:

```
{
 "type": "udp",
 "name": "UDP Stream",
 "url": "udp://192.168.0.57:1234",
 "ts_latency": 200,
 "network_caching": 200,
 "mtu": 0                  // Default 0
}
```

Response Payload

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

3.2 Delete Decode Source

Removes a registered decode source from the pool by UUID index.

Endpoint: `/codec/source/del.json`
Method: `POST` / `GET`

Request Payload

```
{
 "index": "be5018b6abf448aeb8f83716a59c42ac"
}
```

Response Payload

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

3.3 Modify Decode Source

Updates parameters of an existing decode source item.

Endpoint: `/codec/source/modify.json`
Method: `POST` / `GET`

Request Payload

```
{
 "index": "be5018b6abf448aeb8f83716a59c42ac",
 "type": "rtmp",
 "name": "Updated RTMP",
 "url": "rtmp://192.168.0.57",
```

```
 "network_caching": 200,
 "contimeout": 15,
 "rtmp_user": "bbb",
 "rtmp_pwd": "bbb",
 "rtmp_stream_key": "bbb"
}
```

Note: Parameters match Section 3.1 with the addition of the target `index` field. NDI and OMT sources cannot be edited once created.

Response Payload

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

3.4 Get Decode Source List

Queries all registered decode sources and their connectivity status.

Endpoint: `/codec/source/get.json`
Method: `POST` / `GET`

Response Payload

```
{
 "data": [
  {
   "index": "9513222ef9664c0caeec416f2ce13ba1",
   "status": {
    "online": false
   },
   "prop": {
    "index": "9513222ef9664c0caeec416f2ce13ba1",
    "name": "123",
    "type": "rtsp",
    "url": "rtsp://192.168.0.57:554/ch01",
    "transport_protocol": 2,
    "network_caching": 200,
    "rtsp_latency": 2000,
    "rtsp_contimeout": 15,
    "rtsp_user": "",
    "rtsp_pwd": ""
   }
  },
  {
   "index": "20be53d8240647bbbbfb11c70d85a98d",
   "status": {
    "online": false
   },
   "prop": {
    "index": "20be53d8240647bbbbfb11c70d85a98d",
    "name": "ENCODER57 (ndi)",
    "type": "ndi",
    "host": "ENCODER57",
    "channel": "ndi",
    "group": "public",
    "url": "192.168.0.57:5961",
    "network_caching": 0,
    "preview_mode": 0
   }
  }
 ],
 "result": "ok"
}
```

3.5 Select Decode Output Source

Directs a configured decode stream source to physical baseband output (HDMI or SDI).

Endpoint: `/codec/output/select_source.json`
Method: `POST` / `GET`

Request Payload

```
{
 "type": "hdmi",                          // Physical output: "hdmi" or "sdi"
 "index": "9513222ef9664c0caeec416f2ce13ba1"  // UUID from Section 3.4
}
```

Response Payload

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

3.6 Remove / Close Decode Output Source

Disconnects the active decode source from the specified physical output port.

Endpoint: `/codec/output/remove_source.json`
Method: `POST` / `GET`

Request Payload

```
{
 "type": "hdmi" // "hdmi" or "sdi"
}
```

Response Payload

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

3.7 Get Real-Time Decoding Status

Retrieves real-time decoding statistics, resolution, bitrate, and health telemetry for HDMI and SDI outputs.

Endpoint: `/codec/output/get_status.json`
Method: `POST` / `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "hdmi": {
   "status": {
    "online": "connected",
    "resolution": "1920x1080P 60Hz",
    "width": 1920,
    "height": 1080,
    "fps": 60,
    "interlaced": false,
    "video_codec": "H264",
    "video_bits": 7766,
    "audio_codec": "AAC",
    "audio_bits": 129,
    "sample_rate": 48000,
    "channels": 2,
    "no_samples": 1024,
    "reserved": 0
   },
   "prop": {
```

```
      "index": 2,
      "name": "rtsp",
      "type": "rtsp",
      "url": "rtsp://192.168.0.57:554/ch01",
      "network_caching": 200,
      "rtsp_latency": 2000,
      "rtsp_contimeout": 15,
      "transport_protocol": 7,
      "rtsp_user": "a",
      "rtsp_pwd": "b"
    }
   },
   "sdi": {
    "status": {
      "online": "reconnect",
      "resolution": "0x0P 0Hz",
      "width": 0,
      "height": 0,
      "fps": 0,
      "video_codec": "",
      "video_bits": 0,
      "audio_codec": "",
      "audio_bits": 0,
      "sample_rate": 0,
      "channels": 0
    },
    "prop": {
      "index": 12,
      "name": "1111",
      "type": "hls",
      "url": "2223",
      "islive": true,
      "retries": 3,
      "timeout": 15,
      "network_caching": 200,
      "user_id": "",
      "user_pwd": "",
      "user_agent": "",
      "proxy": "",
      "proxy_id": "",
      "proxy_pwd": ""
    }
   }
  }
}
```


3.8 Get Output Interface Configuration

Queries hardware display interface parameters for HDMI and SDI video ports.

Endpoint: `/codec/output/get_output_cfg.json`
Method: `POST` / `GET`

Response Payload


```
{
 "data": {
  "hdmi": {
    "enable": true,
    "intf": "HDMI",
    "sync": 10,                    // Resolution preset value ID (see Section 3.10)
    "format": 4,                   // Color Space value ID (see Section 3.10)
    "range": 0,                    // RGB Range: 0: Auto, 1: LIMITED, 2: FULL
    "sample_rate": 48000,          // Audio sample rate
    "channels": 2,                 // Audio channels count
    "channels_map": [1, 2, 3, 4, 5, 6, 7, 8],
    "gain": 0,                     // Volume gain [-40, 20] dB
    "bind_uvc": false              // Enable USB video output bridge
```

```
  },
  "sdi": {
    "enable": true,
    "intf": "LCD",
    "sync": 10,
    "format": 2,
    "range": 0,
    "sample_rate": 48000,
    "channels": 2,
    "channels_map": [1, 2, 3, 4, 5, 6, 7, 8],
    "gain": 0,
    "bind_uvc": false
  }
 },
 "result": "ok"
}
```

Field Description

Field	Type	Description
enable	Boolean	Port output enable flag (true/false).
sync	Integer	Resolution format preset ID. Values correspond to sync items in Section 3.10.
format	Integer	Color space format ID. Values correspond to format items in Section 3.10.
range	Integer	Color quantization range: 0: Auto, 1: Limited (16-235), 2: Full (0-255).
sample_rate	Integer	Audio sampling frequency in Hz (see Section 3.10).
channels	Integer	Audio channels count.
channels_map	Array	8-channel audio routing map.
gain	Integer	Audio volume gain adjustment in dB [-40, 20].
bind_uvc	Boolean	Whether to bridge decoded stream to USB UVC output.

3.9 Set Output Interface Configuration

Sets display output resolution, color space, audio routing, and UVC bridging for HDMI/SDI.

Endpoint: `/codec/output/set_output_cfg.json`
Method: `POST` / `GET`

Request Payload

```
{
 "type": "hdmi",                    // "hdmi" or "sdi"
 "enable": true,
 "sync": 10,                        // Resolution ID from Section 3.10
 "format": 2,                       // Color space ID from Section 3.10
 "range": 0,                        // 0: Auto, 1: Limited, 2: Full
 "sample_rate": 48000,
 "channels": 2,
 "channels_map": [1, 2, 3, 4, 5, 6, 7, 8],
 "gain": 0,
```

```
 "bind_uvc": false
}
```

Response Payload

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

3.10 Get Supported Output Interface Options & Ranges

Queries all permissible values for color space (format), resolution (sync), RGB quantization range (range), and audio sampling rates (sample_rate) supported by HDMI/SDI.

Endpoint: `/codec/output/get_config_list.json`
Method: `POST` / `GET`

Request Payload

```
{
 "type": "hdmi" // "hdmi" or "sdi"
}
```

Response Payload

```
{
 "data": {
  "format": [
   { "value": 4, "name": "AUTO" },
   { "value": 0, "name": "RGB" },
   { "value": 1, "name": "YUV444" },
   { "value": 2, "name": "YUV422" },
   { "value": 3, "name": "YUV420" }
  ],
  "sync": [
   { "value": 10, "name": "1920x1080P 60Hz",    "width": 1920, "height": 1080, "frameRate": 60,    "interlace": false, "hd": true, "sd":
true },
   { "value": 47, "name": "1920x1080P 59.94Hz", "width": 1920, "height": 1080, "frameRate": 59.94, "interlace": false, "hd": true,
"sd": true },
   { "value": 9,  "name": "1920x1080P 50Hz",    "width": 1920, "height": 1080, "frameRate": 50,    "interlace": false, "hd": true, "sd":
true },
   { "value": 4,  "name": "1920x1080P 30Hz",    "width": 1920, "height": 1080, "frameRate": 30,    "interlace": false, "hd": true, "sd":
true },
   { "value": 48, "name": "1920x1080P 29.97Hz", "width": 1920, "height": 1080, "frameRate": 29.97, "interlace": false, "hd": true,
"sd": true },
   { "value": 3,  "name": "1920x1080P 25Hz",    "width": 1920, "height": 1080, "frameRate": 25,    "interlace": false, "hd": true, "sd":
true },
   { "value": 49, "name": "1920x1080P 23.98Hz", "width": 1920, "height": 1080, "frameRate": 23.98, "interlace": false, "hd": true,
"sd": true },
   { "value": 2,  "name": "1920x1080P 24Hz",    "width": 1920, "height": 1080, "frameRate": 24,    "interlace": false, "hd": true, "sd":
true },
   { "value": 8,  "name": "1920x1080I 60Hz",    "width": 1920, "height": 1080, "frameRate": 60,    "interlace": true,  "hd": true, "sd":
true },
   { "value": 50, "name": "1920x1080I 59.94Hz", "width": 1920, "height": 1080, "frameRate": 59.94, "interlace": true,  "hd": true,
"sd": true },
   { "value": 7,  "name": "1920x1080I 50Hz",    "width": 1920, "height": 1080, "frameRate": 50,    "interlace": true,  "hd": true, "sd":
true },
   { "value": 5,  "name": "1280x720P 50Hz",     "width": 1280, "height": 720,  "frameRate": 50,    "interlace": false, "hd": true, "sd":
true },
   { "value": 51, "name": "1280x720P 59.94Hz",  "width": 1280, "height": 720,  "frameRate": 59.94, "interlace": false, "hd": true, "sd":
true },
   { "value": 6,  "name": "1280x720P 60Hz",     "width": 1280, "height": 720,  "frameRate": 60,    "interlace": false, "hd": true, "sd":
true }
  ],
  "range": [
   { "value": 0, "name": "AUTO" },
   { "value": 1, "name": "LIMITED" },
   { "value": 2, "name": "FULL" }
  ],
```

```
   "sample_rate": [
    { "value": 32000, "name": "32kHz" },
    { "value": 44100, "name": "44.1kHz" },
    { "value": 48000, "name": "48kHz" },
    { "value": 96000, "name": "96kHz" },
    { "value": 0,     "name": "Disable" }
   ]
  },
  "result": "ok"
}
```

3.11 Get Output Audio Volume Levels (dB)

Reads output audio volume levels in decibels for HDMI or SDI ports.

Endpoint: `/codec/output/get_output_real_db.json`
Method: `POST` / `GET`

Request Payload

```
{
  "type": "hdmi" // "hdmi" or "sdi"
}
```

Response Payload

```
{
  "result": "ok",
  "data": [
    -24.039293, // Left channel output dB
    -24.049520  // Right channel output dB
  ]
}
```

3.12 WebRTC Decoding Preview

Initiates low-latency WebRTC video stream monitoring of the current decoder output.

Endpoint: `/codec/webrtc/decode_preview.json`
Method: `POST` / `GET`

Request Payload

```
{
  "mode": 0,          // 0: Video stream mode, 1: Snapshot image mode
  "ip": "192.168.1.1", // Client IP address requesting preview
  "type": "hdmi"       // "hdmi" or "sdi"
}
```

Response Payload

```
{
  "result": "ok",
  "data": {
    "port": 51545,
    "sdp": "v=0\r\no=- 1495799811084970 2 IN IP4 192.168.0.176\r\ns=-\r\nt=0 0\r\na=group:BUNDLE video\r\na=ice-lite
\r\nm=video 9 UDP/TLS/RTP/SAVPF 126\r\nc=IN IP4 0.0.0.0\r\na=ice-ufrag:uoxv\r\na=ice-
pwd:93urglck0zhkz1q4a87clihx\r\na=ice-options:trickle\r\na=fingerprint:sha-256
21:39:6D:25:62:F6:14:A5:28:95:39:DF:7D:F9:DC:D3:13:FA:93:B9:57:54:19:60:62:F9:CF:51:D6:69:DC:D5\r\na=setup:actpass\r\na=
mid:video\r\na=sendonly\r\na=rtcp-mux\r\na=rtpmap:126 H264/90000\r\na=fmtp:126 level-asymmetry-
allowed=1;packetization-mode=1;profile-level-id=42e01f\r\na=candidate:1 1 udp 12345678 192.168.0.176 51545 typ
host\r\na=end-of-candidate\r\n"
  }
}
```

3.13 Get NDI Network Discovery List

Discovers and lists all NDI sources available on the local subnet (mDNS / Discovery Server).

Endpoint: `/codec/discovery/get_result.json`
Method: `POST` / `GET`

Response Payload


```
{
 "result": "ok",
 "data": [
  {
   "device_name": "192168000231",
   "channel_name": "12",
   "name": "192168000231 (12)",
   "url": "192.168.0.226:5961",
   "group": "public"
  },
  {
   "device_name": "192168000231",
   "channel_name": "22222a",
   "name": "192168000231 (22222a)",
   "url": "192.168.0.226:5963",
   "group": "public"
  }
 ]
}
```


3.14 Get OMT Network Discovery List

Discovers and lists all available OMT streaming sources across the local network.

Endpoint: `/codec/discovery/get_omt_result.json`
Method: `POST` / `GET`

Response Payload


```
{
 "result": "ok",
 "data": [
  {
   "device_name": "DESKTOP-JK3MJJQ",
   "channel_name": "vMix - Output 1",
   "name": "DESKTOP-JK3MJJQ (vMix - Output 1)",
   "url": "",
   "group": ""
  },
  {
   "device_name": "PC-202408271729",
   "channel_name": "OBS Outputzhuji",
   "name": "PC-202408271729 (OBS Outputzhuji)",
   "url": "",
   "group": ""
  }
 ]
}
```

4. Recording Module

The Recording Module manages local recording of encoded video/audio streams to attached USB storage drives or SD cards, supporting configurable file splitting, recording tactics, and storage file exploration.

4.1 Get Recording Configuration

Queries current recording configuration profiles for an encoder channel.

Endpoint: `/record/get_config.json`
Method: `POST` / `GET`

Request Payload

```
{
 "id": 1 // 1: encode1, 2: encode2
}
```

Response Payload

```
{
 "result": "ok",
 "data": {
  "id": 1,
  "enable": false,
  "disk_id": "/dev/sda1",
  "dir": "RECORDING",
  "prefix": "REC",
  "format": "ts",
  "split": "time",
  "max_time": 0,
  "max_size": 102400,
  "max_files": 0,
  "tactic": "loop"
 }
}
```

Field Description

Field	Type	Description
id	Integer	Recording task ID (1: encode1, 2: encode2).
dir	String	Target storage directory name.
prefix	String	Recorded file name prefix.
disk_id	String	Target block device path (e.g., /dev/sda1).
format	String	Container format: ts, mp4, mov.
split	String	Segmentation mode: time (by duration).
max_time	Integer	Segment duration in seconds (0 = no limit, effective when split="time").
max_size	Integer	Segment file size in KB (effective when split="size").
max_files	Integer	Maximum file count retention per session (0 = unlimited).
tactic	String	Full disk policy: loop (overwrite oldest

		file) or stop (stop recording when full).
enable	Boolean	Recording running state.

4.2 Get External Storage Device Information

Detects and lists attached external storage devices (USB drives, SD cards) along with filesystem type and space statistics.

Endpoint: `/record/get_all_devices.json`
Method: `POST` / `GET`

Response Payload

```
{
 "result": "ok",
 "data": [
  {
   "device": "/dev/sda1",
   "path": "/mnt/udisk",
   "label": "nhp usb",
   "filesystem": "ntfs",
   "total_size": 29431804,
   "use_size": 2571928,
   "percentage": 9,
   "speed": "--MB/s",
   "remaining_recording_time": 26230
  },
  {
   "device": "/dev/mmcblk1p1",
   "path": "/mnt/sdcard",
   "label": "Removable Disk",
   "filesystem": "exfat",
   "total_size": 249970688,
   "use_size": 4633856,
   "percentage": 2,
   "speed": "--MB/s",
   "remaining_recording_time": -1
  }
 ]
}
```

Field Description

Field	Type	Description
device	String	Linux block device node (e.g., /dev/sda1).
path	String	System mount point directory (e.g., /mnt/udisk).
label	String	Disk volume label name.
filesystem	String	Filesystem type (ntfs, exfat, ext4, fat32).
total_size	Integer	Total capacity in KB.
use_size	Integer	Used space in KB.
percentage	Integer	Utilization percentage (0-100%).
speed	String	Disk write throughput benchmark.

remaining_recording_time	Integer	Estimated remaining recording duration in seconds (-1 indicates unselected device).

4.3 Set Recording Configuration

Configures recording parameters, target disk partition, and file split settings.

Endpoint: `/record/set_config.json`
Method: `POST` / `GET`

Request Payload

```
{
 "id": 1,
 "disk_id": "/dev/sda1", // Device node from Section 4.2
 "format": "ts",       // "ts", "mp4", or "mov"
 "prefix": "REC",
 "dir": "RECORDING",
 "max_time": 600,       // Segment length in seconds (0 = unlimited)
 "tactic": "loop"       // "loop" or "stop"
}
```

Response Payload

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

4.4 Get Recording Status

Queries active recording elapsed time and status.

Endpoint: `/record/get_status.json`
Method: `POST` / `GET`

Request Payload

```
{
 "id": 1
}
```

Response Payload

```
{
 "result": "ok",
 "data": {
  "started": true,
  "start_time": 3202,
  "now_time": 3232
 }
}
```

Calculation: Total recorded duration = now_time - start_time (in seconds).

4.5 Start Recording

Triggers recording initiation for an encoder channel.

Endpoint: `/record/start_record.json`
Method: `POST` / `GET`

Request Payload

```
{
 "id": 1
```

```
}
```

Response Payload

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

4.6 Stop Recording

Terminates recording on the specified channel and finalizes files.

Endpoint: `/record/stop_record.json`
Method: `POST` / `GET`

Request Payload

```
{
 "id": 1
}
```

Response Payload

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

4.7 Get Recorded File List from Device

Retrieves a paginated list of recorded media files on a storage device.

Endpoint: `/record/disk/get_file_from_device.json`
Method: `POST` / `GET`

Request Payload

```
{
 "device": "/dev/sda1",
 "page": 1,
 "limit": 2000
}
```

Response Payload

```
{
 "result": "ok",
 "data": {
  "total": 8,
  "page": 1,
  "limit": 2000,
  "list": [
   {
    "filename": "REC-20260916040857.ts",
    "fpath": "/mnt/udisk/RECORDING/RECORDING/REC-20260916040857.ts",
    "ftype": "file",
    "size": 0,
    "opening": true, // File is actively being written by recorder
    "start_time": "2026-09-16 04:08:57",
    "stop_time": "--",
    "ctime": 1789546137,
    "mtime": 1789546137,
    "atime": 1789546137,
    "block": 0,
    "ino": 89
   },
   {
    "filename": "REC-20260916040541.ts",
    "fpath": "/mnt/udisk/RECORDING/RECORDING/REC-20260916040541.ts",
    "ftype": "file",
    "size": 86116,
```

```
      "opening": false,
      "start_time": "2026-09-16 04:05:41",
      "stop_time": "2026-09-16 04:08:28",
      "ctime": 1789546107,
      "mtime": 1789546107,
      "atime": 1789545941,
      "block": 172232,
      "ino": 86
    }
   ]
  }
}
```


4.8 Eject Storage Device

Safely unmounts and prepares an external storage device for physical removal.

Endpoint: `/record/disk/eject_device.json`
Method: `POST` / `GET`

Request Payload


```
{
 "device": "/dev/sda1"
}
```


Response Payload


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

5. System Module

The System Module controls hardware telemetry, LED indicators, reboot/reset routines, analog audio routing, Type-C interface modes, ZeroTier VPN, SNMP, and SSDP discovery.

5.1 Get Advanced System Settings

Queries configuration status for SSDP, ZeroTier VPN, SNMP, and Type-C USB mode.

Endpoint: `/system/get_advanced_info.json`
Method: `POST` / `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "ssdp": {
   "enable": true,
   "interface": "eth0"
  },
  "zerotier": {
   "enable": false,
   "networkid": ""
  },
  "typec_mode": "rndis",
  "snmp": {
   "enable": false,
   "port": 161,
   "password": "public"
  }
 }
}
```

5.2 Configure ZeroTier VPN

Enables or disables ZeroTier virtual LAN tunnel connection and sets the Network ID.

Endpoint: `/system/zerotier/update_zerotier.json`
Method: `POST` / `GET`

Request Payload

```
{
 "enable": false,
 "networkid": "68bea79acfc41294"
}
```

Response Payload

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

5.3 Configure SNMP Service

Configures the SNMP monitoring daemon port and community password.

Endpoint: `/system/snmp/update_snmp.json`
Method: `POST` / `GET`

Request Payload

```
{
 "enable": false,
 "port": 161,
```

```
 "password": "public"
}
```

Response Payload

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

5.4 Configure SSDP Service

Enables or disables Simple Service Discovery Protocol (SSDP) and specifies binding interface.

Endpoint: `/system/ssdp/update_ssdp.json`
Method: `POST` / `GET`

Request Payload

```
{
 "enable": false,
 "interface": "eth0" // "eth0" or "eth1"
}
```

Response Payload

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

5.5 Set USB Type-C Operating Mode

Switches the USB Type-C port between RNDIS (Ethernet over USB) and UVC (USB Video Class webcam).

Endpoint: `/system/set_typec_mode.json`
Method: `POST` / `GET`

Request Payload

```
{
 "usbc_mode": "rndis" // "rndis" (USB Network) or "uvc" (USB Webcam)
}
```

Response Payload

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

Important: Changing the Type-C operating mode requires a device reboot to take effect.

5.6 Get Analog Audio Output Routing

Reads current routing settings for 3.5mm Line Out and Headphone audio jacks.

Endpoint: `/codec/get_analog.json`
Method: `POST` / `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "lineout": 0,
  "headphone": 0
 }
}
```

5.7 Set Analog Audio Output Routing

Routes audio sources to the Line Out and Headphone physical jacks.

Endpoint: `/codec/set_analog.json`
Method: `POST` / `GET`

Request Payload

```
{
 "lineout": 1,
 "headphone": 0
}
```

Value Reference Table

Mode	Value	Line Out / Headphone Source
Encoding Mode	0	None / Mute
	1	Line In
	2	Encode Channel 1 Audio
	3	Encode Channel 2 Audio
Decoding Mode	0	None / Mute
	1	Line In
	2	HDMI Audio
	3	SDI Audio

Response Payload

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

5.8 Configure NDI Discovery Server

Sets the IP address of an external centralized NDI Discovery Server.

Endpoint: `/codec/discovery/set_discovery_server.json`
Method: `POST` / `GET`

Request Payload

```
{
 "enable_server": true,
 "server": "192.168.0.91"
}
```

Response Payload

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

5.9 Get NDI Discovery Server Configuration

Queries the configured NDI Discovery Server settings.

Endpoint: `/codec/discovery/get_discovery_server.json`

Method: `POST` / `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "enable_server": true,
  "server": "192.168.0.91"
 }
}
```

5.10 Reboot Device

Executes an immediate operating system reboot on the hardware.

Endpoint: `/system/reboot.json`
Method: `GET`

Response Payload

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

5.11 Restore Factory Defaults

Resets the device configuration to factory default state and reboots.

Endpoint: `/system/refactory.json`
Method: `GET`

Response Payload

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

5.12 Get LED Status & Configuration

Reads physical LED lamp indicators (Tally, Recording, Streaming, Find Device) and VU volume meter status.

Endpoint: `/led/get_leds.json`
Method: `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "enable": true,
  "tally": {
   "program": 1,
   "preview": 2,
   "brightness": 100
  },
  "record": {
   "enable": true,
   "color": 1,
   "brightness": 100,
   "encode_id": 1
  },
  "stream": {
   "enable": true,
   "color": 2,
   "brightness": 100,
   "encode_id": 1
```

```
  },
  "find_device": {
    "enable": false,
    "color": 1,
    "brightness": 100,
    "mode": 1,
    "duration": 5
  },
  "vu": {
    "enable": true,
    "encode_id": 1,
    "decode_id": 1,
    "id": 1
  }
 }
}
```


Parameter Reference

Parameter	Description
enable	Global LED hardware enable switch.
*.color	Color code: 0: None (Off), 1: Red, 2: Green, 3: Blue, 4: White.
*.brightness	LED brightness percentage: [0, 100].
find_device.mode	Flashing animation: 1: Flashlight, 2: Attention strobe, 3: Pulsing glow.
find_device.duration	Auto-shutoff timer in minutes (0 = run indefinitely).
vu.id	In Encoding mode: 1: encode1, 2: encode2. In Decoding mode: 1: hdmi, 2: sdi.
vu.enable	Audio LED volume meter bar graph toggle.

5.13 Set LED Status & Configuration

Sets hardware LED brightness, color, and operating behaviors.

Endpoint: `/led/set_leds.json`
Method: `POST`

Request Payload


```
{
 "enable": false,
 "find_device": {
   "enable": true,
   "color": 2,
   "mode": 1
 },
 "tally": {
   "program": 1,
   "preview": 1,
   "brightness": 75
 },
 "record": {
   "enable": true,
   "color": 2,
   "brightness": 58,
   "encode_id": 1
 },
 "stream": {
   "enable": true,
```

```
  "color": 3,
  "brightness": 74,
  "encode_id": 1
 },
 "vu": {
  "enable": true,
  "id": 1
 }
}
```

Response Payload

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

5.14 Get Real-Time System Performance Status

Reads runtime hardware load: uptime, total memory, CPU utilization, and memory usage.

Endpoint: `/system/get_system_state.json`
Method: `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "ptime": 7186,      // System uptime in seconds
  "memTotal": 1956,   // Total system RAM in MB
  "cpu": 8,           // CPU load percentage (0-100%)
  "mem": 32           // RAM utilization percentage (0-100%)
 }
}
```

5.15 Get System Hardware & Firmware Information

Returns serial number, hardware revision, software build version, and active system operating mode.

Endpoint: `/system/get_system_info.json`
Method: `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "DEVICE_NAME": "V410-V4105464864646",
  "DEVICE_TYPE": "V410",
  "HARDWARE_VERSION": "2.2",
  "SOFTWARE_VERSION": "V1.00.0033",
  "BUILD_INFO": "20260720_171602",
  "SERIAL": "V4105464864646",
  "mode": "ENCODING" // "ENCODING" or "DECODING"
 }
}
```

6. Network Module

The Network Module manages Ethernet interfaces (LAN1, LAN2), static/DHCP IP configuration, USB tethered networking, and Wi-Fi scanning/connections.

6.1 Get Network Link Configuration

Queries status, IP configuration, gateway, DNS, and traffic throughput for all network interfaces.

Endpoint: `/network/get_link_list.json`
Method: `POST` / `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "eth": [
   {
    "intf": "LAN1",
    "interface": "eth0",
    "type": "eth",
    "running": true,              // Physical link connected (cable plugged)
    "method": "static",           // "dhcp" or "static"
    "ip": "192.168.0.232",
    "netmask": "255.255.252.0",
    "gateway": "192.168.1.1",
    "dns": ["8.8.8.8", "114.114.114.114"],
    "mac": "72:2C:4A:1B:4F:0A",
    "ipv6": "fe80::702c:4aff:fe1b:4f0a",
    "ipv6_scope": "Link",
    "up": 226,                    // Upload throughput in Kbps
    "down": 8979                  // Download throughput in Kbps
   },
   {
    "intf": "LAN2",
    "interface": "eth1",
    "type": "eth",
    "running": false,
    "method": "static",
    "mac": "76:2C:4A:1B:4F:0A",
    "dns": [],
    "up": 0,
    "down": 0
   }
  ],
  "wlan": [
   {
    "intf": "Wi-Fi",
    "interface": "wlan0",
    "type": "wlan",
    "up": 0,
    "down": 0,
    "dns": []
   }
  ]
 }
}
```

6.2 Configure Network Connection

Configures IP addressing mode (DHCP or Static), subnet mask, gateway, and DNS servers.

Endpoint: `/network/set_link_connection.json`
Method: `POST` / `GET`

Request Payload

```
{
 "interface": "eth0",        // "eth0" or "eth1"
 "method": "static",         // "dhcp" or "static"
 "ip": "192.168.0.232",      // Ignored when method="dhcp"
 "netmask": "255.255.252.0",
 "gateway": "192.168.1.1",
 "dns": ["192.168.1.1", ""]
}
```

Important: If the IP address is altered, the existing connection immediately closes and no HTTP response payload is returned.

6.3 Get USB Network Interface Information

Queries status of the USB tethered network adapter (usb0).

Endpoint: `/network/get_usb_net.json`
Method: `POST` / `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "interface": "usb0",
  "type": "usb",
  "ip": "192.168.64.74",
  "netmask": "255.255.255.0",
  "mac": "22:D5:C5:D0:F9:6E",
  "enable": true,
  "connected": false
 }
}
```

6.4 Set USB Network IP Address

Assigns a static IP address to the USB network interface.

Endpoint: `/network/set_usb_net.json`
Method: `POST` / `GET`

Request Payload

```
{
 "ip": "192.168.91.10"
}
```

Response Payload

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

6.5 Get Wi-Fi Status

Queries whether the wireless Wi-Fi radio is enabled.

Endpoint: `/network/wifi/get_config.json`
Method: `POST` / `GET`

Response Payload

```
{
 "result": "ok",
 "data": {
  "enable": true
```

```
 }
}
```

6.6 Enable / Disable Wi-Fi

Powers the wireless Wi-Fi module on or off.

Endpoint: `/network/wifi/enable_wifi.json`
Method: `POST` / `GET`

Request Payload

```
{
 "enable": true
}
```

Response Payload

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

6.7 Get Scanned Wi-Fi Networks List

Scans for nearby wireless access points and returns signal strengths and security flags.

Endpoint: `/network/wifi/get_list.json`
Method: `POST` / `GET`

Response Payload

```
{
 "result": "ok",
 "data": [
  {
   "ssid": "TP-LINK_2C15",
   "realname": "TP-LINK_2C15",
   "mac": "9c:47:82:a2:2c:15",
   "signal": "-35",          // RSSI in dBm
   "flags": "sae",           // WPA3 / SAE security
   "connect": false,
   "generation": 0
  },
  {
   "ssid": "G_5G",
   "realname": "G_5G",
   "mac": "08:9b:4b:b9:1e:66",
   "signal": "-48",
   "flags": "wpa",           // WPA2 / WPA security
   "connect": false,
   "generation": 0
  }
 ]
}
```

6.8 Connect to Wi-Fi Network

Associates and authenticates with a discovered Wi-Fi network.

Endpoint: `/network/wifi/connect_wifi.json`
Method: `POST` / `GET`

Request Payload

```
{
 "ssid": "%40Ruijie-sC214",          // URL-encoded SSID name
 "mac": "c2%3Aa4%3A76%3A91%3Ac2%3A16", // URL-encoded BSSID MAC address
 "psk": "password123"                // Pre-shared Wi-Fi password
```

```
}
```

Response Payload

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

6.9 Connect to Hidden Wi-Fi Network

Connects to an unbroadcasted / hidden Wi-Fi network.

Endpoint: `/network/wifi/add_hidden_wifi.json`
Method: `POST` / `GET`

Request Payload

```
{
 "ssid": "%40Ruijie-sC214", // URL-encoded SSID
 "psk": "password123",
 "flags": "wpa"            // Security protocol (e.g., "wpa")
}
```

Response Payload

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

6.10 Disconnect / Forget Wi-Fi Network

Disconnects from an active Wi-Fi connection and removes stored credentials.

Endpoint: `/network/wifi/disconnect_wifi.json`
Method: `POST` / `GET`

Request Payload

```
{
 "ssid": "%40Ruijie-sC214",
 "mac": "c2%3Aa4%3A76%3A91%3Ac2%3A16"
}
```

Response Payload

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

7. Scheduled Tasks Module

The Scheduled Tasks Module provides an automated cron scheduler on the device to trigger streaming, recording, or scheduled system reboots at predefined times or calendar recurrences.

7.1 Get Scheduled Tasks

Queries all configured cron tasks filtered by type.

Endpoint: `/scheduler/get.json`

Method: `POST`

Request Payload

```
{
 "type": 1,      // Task type: 1: Streaming, 2: Recording, 3: Reboot
 "encode_id": ""  // Target encoder ID for recording tasks
}
```

Response Payload

```
{
 "result": "ok",
 "data": [
  {
   "index": 1,
   "prop": {
    "title": "New Scheduler",
    "desc": "",
    "type": 1,
    "stream_id": "2",
    "encode_id": 1,
    "enable": true,
    "start_time": "18:10:05",
    "end_time": "18:12:10",
    "repeat_event": 1,
    "repeat_week_day": [0, 0, 0, 0, 0, 0, 0],
    "repeat_month_week": 1,
    "run_state": 1
   }
  }
 ]
}
```

Field Description

Field	Type	Description
title	String	Task name.
desc	String	Task notes / description.
type	Integer	Task category: 1: Streaming, 2: Recording, 3: Reboot.
stream_id	String	Streaming service ID associated with this task.
encode_id	Integer	Target encoder channel ID.
start_time	String	Start time formatted as hh:mm:ss.
end_time	String	End time formatted as hh:mm:ss. If end_time is earlier than start_time, the task spans overnight into the next day.

		Cannot be identical to start_time.
repeat_event	Integer	Recurrence rule: 1: Daily, 2: Weekly, 3: Monthly, 4: One-time execution.
repeat_week_day	Array	7-element boolean array for Monday through Sunday (1: active, 0: inactive).
repeat_month_week	Integer	Week of month ordinal [1, 5] (effective when repeat_event=3).
run_state	Integer	Runtime state: 1: Waiting / Pending, 2: Running, 3: Execution Failed, 4: Expired / Inactive.
enable	Boolean	Task enabled flag.

7.2 Create Scheduled Task

Registers a new scheduled streaming or recording automation task.

Endpoint: `/scheduler/create.json`
Method: `POST`

Request Payload

```
{
 "type": 1,                    // 1: Streaming, 2: Recording (Reboot tasks cannot be created, only edited)
 "title": "Automated Broadcast",
 "desc": "Morning live stream event",
 "stream_id": "2",
 "encode_id": 1,
 "start_time": "08:00:00",
 "end_time": "10:00:00",
 "repeat_event": 2,              // 1: Daily, 2: Weekly, 3: Monthly, 4: One-time
 "repeat_week_day": [1, 1, 1, 1, 1, 0, 0], // Monday through Friday
 "repeat_month_week": 1,
 "start_date": "2026-10-01"          // Date string for one-time tasks
}
```

Note: System Reboot tasks cannot be newly created; they can only be modified via Section 7.3.

Response Payload

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

7.3 Update Scheduled Task

Modifies an existing scheduled automation task by its index.

Endpoint: `/scheduler/update.json`
Method: `POST`

Request Payload

```
{
 "index": 1,                    // Task index from Section 7.1
 "title": "Updated Broadcast",
 "desc": "",
 "stream_id": "2",
 "encode_id": 1,
 "start_time": "18:10:05",
 "end_time": "18:12:10",
```

```
 "repeat_event": 1,
 "repeat_week_day": [0, 0, 0, 0, 0, 0, 0],
 "repeat_month_week": 1,
 "start_date": "2026-12-03"
}
```

Response Payload

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

7.4 Enable / Disable Scheduled Task

Toggles an automated schedule task active or inactive.

Endpoint: `/scheduler/enable.json`
Method: `POST`

Request Payload

```
{
 "index": 1,
 "enable": false
}
```

Response Payload

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

7.5 Delete Scheduled Task

Deletes a scheduled task definition.

Endpoint: `/scheduler/delete.json`
Method: `POST` / `GET`

Request Payload

```
{
 "index": 1
}
```

Response Payload

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

8. User Management Module

The User Management Module provides endpoints for account authentication, password management, and Two-Factor Authentication (2FA / TOTP) configuration.

8.1 User Login

Authenticates user credentials and checks two-factor authentication (2FA) enforcement.

Endpoint: `/system/user/login.json`
Method: `POST` / `GET`

Request Payload

```
{
 "username": "admin",
 "password": "password123"
}
```

Response Payload

```
{
 "result": "ok",
 "data": {
  "username": "admin",
  "role": "admin",
  "AuthToken": "b8c8169af00d4600e6304e6ad5ecb6c0",
  "changed": true,
  "enable_2fa": true,    // Defaults to false if omitted
  "activate_2fa": false  // Defaults to false if omitted
 }
}
```

8.2 Toggle Two-Factor Authentication (2FA)

Enables or disables Two-Factor Authentication requirements for a user account.

Endpoint: `/system/user/enable_2fa.json`
Method: `POST` / `GET`

Request Payload

```
{
 "username": "admin",
 "enable_2fa": true
}
```

Response Payload

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

8.3 Activate Two-Factor Authentication (2FA)

Verifies a 6-digit TOTP verification code from an authenticator application to finalize 2FA activation.

Endpoint: `/system/user/activate_2fa.json`
Method: `POST` / `GET`

Request Payload

```
{
 "username": "admin",
 "code": "112321" // 6-digit TOTP code
}
```

Response Payload

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

8.4 Get User Status

Queries a user's 2FA enrollment status and TOTP secret key for QR code binding.

Endpoint: `/system/user/get_user_status.json`
Method: `POST` / `GET`

Request Payload

```
{
 "username": "admin"
}
```

Response Payload

```
{
 "result": "ok",
 "data": {
   "username": "admin",
   "enable_2fa": true,
   "activate_2fa": false,
   "token_2fa": "GVKEKN2KG5KE4NCMJJDU2V2QLBBVQRBVINDECS2EJJFFAUKUGNGA" // Base32 TOTP secret string
 }
}
```

Appendix: Error Codes & Status Reference

Standard Error Response Format

When an API request fails, the server responds with a JSON payload containing an integer `error` code and `result: "error"`:

```
{
 "error": 45073,
 "result": "error"
}
```

Error Codes Lookup Table

Decimal Code	Hex Code	Error Message	Description & Resolution
0	0x0000	There was an error	Generic system error. Check system logs.
1	0x0001	Invalid Parameter	Request payload contains invalid, out-of-range, or missing parameters.
2	0x0002	Request failed	Server failed to execute the requested command.
3	0x0003	User information timed-out	Session token has expired or is invalid. Re-authenticate via login API.
4	0x0004	System call failed	Underlying Linux OS or kernel system call returned an error.
5	0x0005	Invalid request	Malformed HTTP request or unsupported method.
40961	0xA001	OSD numbers maximum	Maximum number of concurrent OSD overlay layers reached.
40962	0xA002	OSD modification failed	Failed to update the specified OSD overlay layer.
40963	0xA003	Volume modification failed	Failed to adjust audio gain or volume level.
40964	0xA004	The audio encoding is abnormal, please try again later	Audio DSP or encoding pipeline is temporarily busy or recovering.
45057	0xB001	No such stream type	Unsupported streaming protocol specified.
45058	0xB002	The number of streams reaches the upper limit	Maximum number of active streaming services reached.
45059	0xB003	Port is occupied	Target network port is already in use by another service.
45060	0xB004	Recording not exist	Specified recording task or session ID does not exist.
45061	0xB005	Disk not exist	Target storage device or mount path not found.
45062	0xB006	Recording failed	Recorder failed to open file or

			write stream data to disk.
45063	0xB007	No preview information	Preview stream is unavailable or encoder is stopped.
45064	0xB008	No corresponding encoding information	Encoder channel ID is not active or uninitialized.
45065	0xB009	Invalid IP information	Provided IP address format is invalid.
45067	0xB00B	The server already exists	A stream service with identical binding or port already exists.
45068	0xB00C	Only one NDI®|HX2 streaming is allowed	Concurrency limit: Only 1 NDI®|HX2 egress stream can run at a time.
45069	0xB00D	Only one NDI®|HX3 streaming is allowed	Concurrency limit: Only 1 NDI®|HX3 egress stream can run at a time.
45070	0xB00E	Only one HLS streaming is allowed	Concurrency limit: Only 1 HLS egress stream can run at a time.
45071	0xB00F	Only one RTSP streaming is allowed	Concurrency limit: Only 1 RTSP server can run at a time.
45072	0xB010	Only one SRT Listener streaming is allowed	Concurrency limit: Only 1 SRT in Listener mode is permitted.
45073	0xB011	NDI®|HX3/NDI HB/OMT and other streaming protocols cannot be enabled simultaneously	Protocol mutual exclusion rule violated.
45074	0xB012	Stream simultaneously to 8 servers at most	Hardware stream output limit reached (maximum 8 concurrent targets).
45075	0xB013	NDI®|HX3/NDI HB/OMT and recording cannot be enabled simultaneously	High-bitrate encoding cannot operate concurrently with local recording.
45076	0xB014	Limited to a single encoding stream for 4K	4K UHD processing limits the device to a single active encoder channel.
45077	0xB015	Allowed to decode one NDI or OMT	Decoding limit: Only 1 NDI or OMT source can be decoded simultaneously.
45078	0xB016	Only one NDI HB streaming is allowed	Concurrency limit: Only 1 Full NDI (High Bandwidth) egress stream allowed.
45079	0xB017	Only one OMT streaming is allowed	Concurrency limit: Only 1 OMT stream service can be enabled.
45080	0xB018	The channel name already exists	NDI or OMT channel name collision detected on local network.

45081	0xB019	OMT and NDI HB resolution up to 1920x1080	Full NDI HB and OMT are constrained to a maximum resolution of 1080p.
45088	0xB020	UVC mode is already enabled	USB Type-C port is already assigned to UVC webcam mode.
45089	0xB021	Cannot enable UVC while RNDIS mode is active	Conflicting USB modes. Must disable RNDIS or reboot to change mode.
49153	0xC001	Exceed the image limit	Maximum number of stored OSD / background images exceeded.
49154	0xC002	Image upload failed	File transfer corrupted or rejected by server.
49155	0xC003	Unable to delete the current recording file	File is currently locked and being recorded into.
49156	0xC004	Unable to add image with the same name	An image file with the identical filename already exists in storage.
49157	0xC005	The device is currently in use	Hardware resource or storage drive is locked by an active process.
49158	0xC006	The disk is full and recording cannot be started	Target storage partition has zero remaining free space.
53249	0xD001	Wrong Password	Authentication failed due to incorrect password.
53250	0xD002	Account doesn't exist	Specified username does not exist on the device.
53251	0xD003	The account already exists and cannot be added repeatedly	Username already registered.
53252	0xD004	Unable to delete administrator	Root / default administrator account cannot be deleted.
53253	0xD005	Failed to set time zone	Invalid time zone identifier or system clock adjustment failure.
53254	0xD006	Invalid Firmware	Firmware package file corrupted or unsigned.
53255	0xD007	Invalid firmware type	Firmware package is not compatible with this device model.
53256	0xD008	Invalid firmware version	Firmware package version is invalid or downgrade disallowed.
53257	0xD009	Invalid firmware path	Specified upgrade file path does not exist on filesystem.
53258	0xD00A	Upload failed	Firmware upload stream disconnected prematurely.

61441	0xF001	Shortcuts Key Conflict	Front panel shortcut button assignment collision.
61442	0xF002	Nginx check error	Web reverse proxy configuration test failed.
61696	0xF100	Numbers maximum	Reached maximum allowed number of scheduled task entries.
61697	0xF101	Operation failed. Please disable the scheduled task first	Task cannot be edited or deleted while actively running. Disable first.

End of MIRI Device RESTful API Specification.