Muximus API Documentation

MuxLab Part # 500813 / API Document v1.5

INTRODUCTION

The Muximus API allows you to control your Muximus Network Controller for integration with third party tools.

Note: for third-party CMS application development, advanced APIs are available for further management of the device. Please contact support@muxlab.com if you require such APIs.

CONNECTING TO THE API - HTTP METHOD

Connecting to the API is as simple as including the Bearer Token header in your HTTP requests. You can obtain your Bearer Token in Muximus by logging in, under Profile > Password > API Key. The header would look like this:

```
Authorization: Bearer <my-token-here>
```

Please note, each user in Muximus has a separate API key.

You can find relevant information about HTTP Authentication Headers over here: `https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication`.

The API is called via the unit's IP address, using port 5001 as follows:

```
https://[device-ip]:5001/[endpoint]
```

Unless specified otherwise, the `Content-Type` is `application/json`

The APIs are detailed in the sections that follow.

API Endpoints - HTTP METHOD

Presets: *A preset can include connections for RX-TX, Video Wall, Multiview, Signage, and Dante. A preset can specify the connection type (audio, video, usb, etc) as well as streaming protocols (multicast, RTSP etc).*

```
Method: POST

Path: /presets/{PRESET_ID}/activate
```

Description: Activate a Preset that was created in the "Presets & Routing" page. The preset ID can be found in the 3-dots (Kebab) ⋮ menu of a saved preset.

Controls: *A control can include a command for RS-232, IR, and CEC.*

```
Method: POST

Path: /controls/{CONTROL_ID}/activate
```

Description: Run a Control that was created in the "Controls" page. The Control ID can be found in the 3-dots (Kebab) ⋮ menu in the controls list page.

```
Method: POST

Path: /devices/ send-data-rs232-by-device-id

Body: {

    "device_id": "DEVICE-ID",

    "hexdata": "hexadecimal data"

}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

```
Method: POST

Path: /devices/ send-data-ir-by-device-id

Body: {

    "device_id": "DEVICE-ID",
```

```
    "hexdata": "hexadecimal data"
}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

```
Method: POST
Path: /devices/ send-cec-command-by-device-id
Body: {
    "device_id": "DEVICE-ID",
    "command": "powerOn"
}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

`command` can be any of these values: 'standBy', 'powerOn', 'volumeUp', 'volumeDown', 'volumeMute'.

Connections: *Connect receivers, transmitters, multiviews and videowalls*

For the commands below that specify a Transmitter ID, you can use `00-00-00-00-00-00` to STOP the video playback.

Transmitter to Receiver

```
Method: POST
Path: /devices/connect
Body: {
    "presetConnections": [
        {
            "receiver_uuid": "RX-ID",
            "transmitter_uuid": "TX-ID",
            "connectionType": "all"
        }
    ]
```

```
}
```

Description: The Device ID can be found in the 3-dots (Kebab) menu in the Device Manager page.

`connectionType` can be any of these values, according to what the device in question supports: 'all', 'audio-video', 'audio', 'video', 'usb', 'rs232', 'ir', 'gpio'

```
Method: POST

Path: /devices/connect-direct

Body: {

    "receiver_uuid": "RX-ID",

    "transmitter_uuid": "TX-ID"

}
```

Description: The Device ID can be found in the 3-dots (Kebab) menu in the Device Manager page.

`connectionType` is optional, 'all' by default and can be any of these values, according to what the device in question supports: 'all', 'audio-video', 'audio', 'video', 'usb', 'rs232', 'ir', 'gpio', 'rs485', 'relay'.

`Protocol` is optional, 'auto' by default and can be any of these values, according to what the device in question supports: 'auto', 'rtsp', 'hls', 'ts', 'flv', 'multicast'

Multiview to Receiver

```
Method: POST

Path: /multiviews/connect

Body: {

    "multiviewPresetConnections": [

        {

            "multiview_uuid": "MULTIVIEW-ID",

            "receiver_uuid": "RX-ID",

            "receiver_type": "rx"

        }
```

```
    ]
}
```

Description: The Device ID can be found in the 3-dots (Kebab)  menu in the Device Manager page. The Multiview ID can be found in the Kebab menu in the Multiviews page.

Video Wall Connections

```
Method: POST

Path: /videoWalls/connect

Body: {

    "videoWallPresetConnections": [

        {

            "videoWall_uuid": "VIDEOWALL-ID",

            "videoWall_layout_uuid": "VIDEOWALL-LAYOUT-ID",

            "region_connections": [

                {

                    "region_id": "ZONE-ID",

                    "transmitter_uuid": "TX-ID or MULTIVIEW-ID",

                    "source_type": "multiview" or "tx"

                }

            ]

        }

    ]

}
```

Description: The receiver or transmitter IDs can be found in the 3-dots (Kebab)  menu in the Device Manager page. The Multiview ID can be found in the Kebab menu in the Multiviews page. The video wall ID can be found in the kebab menu in the video walls page.

When connecting a multiview source, use the multiview ID in the `transmitter_uuid` field, and set the `source_type` to `"multiview"`. Otherwise, enter the transmitter ID in the `transmitter_uuid field` and set the `source_type` to `"tx"`.

`videoWallPresetConnections` is an array, so you can send multiple connections to multiple video walls in one command. Similarly, `region_connections` is an array, so you can send multiple ZONE connections to the same video wall.

500890 Signage Commands: Send AV over IP content to your 500890 Signage player, either to the whole screen, or to a window (zone) on the screen.

Using a transmitter as a source

```
Method: POST

Path: /signage/connect

Body: {

    "presetConnections": [

        {

            "receiver_uuid": "RX-ID (500890 model)",

            "transmitter_uuid": "TX-ID",

            "window_id": "all", or "1" to "6",

            "protocol": "auto", or "RTSP", "TS", "MULTICAST"

        }

    ]

}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

`connectionType` can be any of these values, according to what the device in question supports: 'all', 'audio-video', 'audio', 'video', 'usb', 'rs232', 'ir', 'gpio'

Using a multiview as a source, full screen only (cannot specify a window).

```
Method: POST
```

```
Path: /multiviews/connect

Body: {

    "multiviewPresetConnections": [

        {

            "multiview_uuid": "MULTIVIEW-ID",

            "receiver_uuid": "RX-ID (500890 model)",

            "receiver_type": "signage"

        }

    ]

}
```

Description: The Device ID can be found in the 3-dots (Kebab)  menu in the Device Manager page.

CONNECTING TO THE API - TCP or UDP METHOD

Connecting to the API is as simple as sending a TCP or UDP command with an authentication token (Bearer Token). You can obtain your Bearer Token in Muximus by logging in, under Profile > Password > API Key.

The TCP port is **9000**.

The UDP port is **9500**.

The command would look like this:

```
IP: Unit IP
Port: 9000 (TCP) or 9500 (UDP)
Command:
{
      "path": "ENDPOINT_PATH",
      "method": "POST" or "GET",
      "bearerToken": "BEARER_TOKEN",
      "body": if applicable
}
```

Please note, each user in Muximus has a separate API key.

The APIs are detailed in the sections that follow.

API Endpoints - TCP or UDP METHOD

Presets: *A preset can include connections for RX-TX, Video Wall, Multiview, Signage, and Dante. A preset can specify the connection type (audio, video, usb, etc) as well as streaming protocols (multicast, RTSP etc).*

```
{
      "path": "/presets/{PRESET_ID}/activate",
      "method": "POST",
      "bearerToken": "YOUR_TOKEN_HERE",
      "body": {}
}
```

Description: Activate a Preset that was created in the "Presets & Routing" page. The preset ID can be found in the 3-dots (Kebab) menu of a saved preset.

Controls: *A control can include a command for RS-232, IR, and CEC.*

```
{
      "path": "/controls/{CONTROL_ID}/activate",
      "method": "POST",
      "bearerToken": "YOUR_TOKEN_HERE",
      "body": {}
}
```

Description: Run a Control that was created in the "Controls" page. The Control ID can be found in the 3-dots (Kebab) ⋮ menu in the controls list page.

```
{
      "path": "/devices/send-data-rs232-by-device-id",
      "method": "POST",
      "bearerToken": "YOUR_TOKEN_HERE",
      "body": {
         "device_id": "DEVICE-ID",
         "hexdata": "hexadecimal data"
      }
}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

```
{
      "path": "/devices/send-data-ir-by-device-id",
      "method": "POST",
      "bearerToken": "YOUR_TOKEN_HERE",
      "body": {
         "device_id": "DEVICE-ID",
```

```
        "hexdata": "hexadecimal data"
    }
}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

```
{
    "path": "/devices/send-cec-command-by-device-id",
    "method": "POST",
    "bearerToken": "YOUR_TOKEN_HERE",
    "body": {
        "device_id": "DEVICE-ID",
        "command": "powerOn"
    }
}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

`command` can be any of these values: 'standBy', 'powerOn', 'volumeUp', 'volumeDown', 'volumeMute'.

Connections: *Connect receivers, transmitters, multiviews and videowalls*

For the commands below that specify a Transmitter ID, you can use `00-00-00-00-00-00` to STOP the video playback.

Transmitter to Receiver

```
{
    "path": "/devices/connect",
    "method": "POST",
    "bearerToken": "YOUR_TOKEN_HERE",
    "body": {
     "presetConnections": [
```

```
        {
            "receiver_uuid": "RX-ID",
            "transmitter_uuid": "TX-ID",
            "connectionType": "all"
        }
      ]
    }
}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

`connectionType` can be any of these values, according to what the device in question supports: 'all', 'audio-video', 'audio', 'video', 'usb', 'rs232', 'ir', 'gpio'

```
{
    "path": "/devices/connect-direct",
    "method": "POST",
    "bearerToken": "YOUR_TOKEN_HERE",
    "body": {
      "receiver_uuid": "RX-ID",
      "transmitter_uuid": "TX-ID"
    }
}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

`connectionType` is optional, 'all' by default and can be any of these values, according to what the device in question supports: 'all', 'audio-video', 'audio', 'video', 'usb', 'rs232', 'ir', 'gpio', 'rs485', 'relay'.

`Protocol` is optional, 'auto' by default and can be any of these values, according to what the device in question supports: 'auto', 'rtsp', 'hls', 'ts', 'flv', 'multicast'.

Multiview to Receiver

```
{
```

```
    "path": "/multiviews/connect",
    "method": "POST",
    "bearerToken": "YOUR_TOKEN_HERE",
    "body": {
        "multiviewPresetConnections": [
            {
                "multiview_uuid": "MULTIVIEW-ID",
                "receiver_uuid": "RX-ID",
                "receiver_type": "rx"
            }
        ]
    }
}
```

Description: The Device ID can be found in the 3-dots (Kebab)  menu in the Device Manager page. The Multiview ID can be found in the Kebab menu in the Multiviews page.

Video Wall Connections

```
{
    "path": "/videoWalls/connect ",
    "method": "POST",
    "bearerToken": "YOUR_TOKEN_HERE",
    "body": {
        "videoWallPresetConnections": [
            {
                "videoWall_uuid": "VIDEOWALL-ID",
```

```
                "videoWall_layout_uuid": "VIDEOWALL-LAYOUT-ID",
                "region_connections": [
                    {
                        "region_id": "ZONE-ID",
                    "transmitter_uuid": "TX-ID or MULTIVIEW-ID",
                    "source_type": "multiview" or "tx"
                      }
                ]
            }
        ]
    }
}
```

Description: The receiver or transmitter IDs can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page. The Multiview ID can be found in the Kebab menu in the Multiviews page. The video wall ID can be found in the kebab menu in the video walls page.

When connecting a multiview source, use the multiview ID in the `transmitter_uuid` field, and set the `source_type` to `"multiview"`. Otherwise, enter the transmitter ID in the `transmitter_uuid field` and set the `source_type` to `"tx"`.

`videoWallPresetConnections` is an array, so you can send multiple connections to multiple video walls in one command. Similarly, `region_connections` is an array, so you can send multiple ZONE connections to the same video wall.

500890 Signage Commands: Send AV over IP content to your 500890 Signage player, either to the whole screen, or to a window (zone) on the screen.

Using a transmitter as a source

```
{
    "path": "/signage/connect ",
    "method": "POST",
    "bearerToken": "YOUR_TOKEN_HERE",
```

```
      "body": {
            "presetConnections": [
               {
                  "receiver_uuid": "RX-ID (500890 model)",
                   "transmitter_uuid": "TX-ID",
                   "window_id": "all", or "1" to "6",
                   "protocol": "auto", or "RTSP", "TS", "MULTICAST"
               }
          ]
      }
}
```

Description: The Device ID can be found in the 3-dots (Kebab) ⋮ menu in the Device Manager page.

`connectionType` can be any of these values, according to what the device in question supports: 'all', 'audio-video', 'audio', 'video', 'usb', 'rs232', 'ir', 'gpio'

Using a multiview as a source, full screen only (cannot specify a window).

```
{
      "path": "/multiviews/connect ",
      "method": "POST",
      "bearerToken": "YOUR_TOKEN_HERE",
      "body": {
          "multiviewPresetConnections": [
              {
                   "multiview_uuid": "MULTIVIEW-ID",
                   "receiver_uuid": "RX-ID (500890 model)",
                   "receiver_type": "signage"
               }
```

```
            ]

        }

}
```

SUPPORT

For support in integrating these APIs, please send a request to support@muxlab.com and you will put in touch with someone who can assist you with your API integrations.