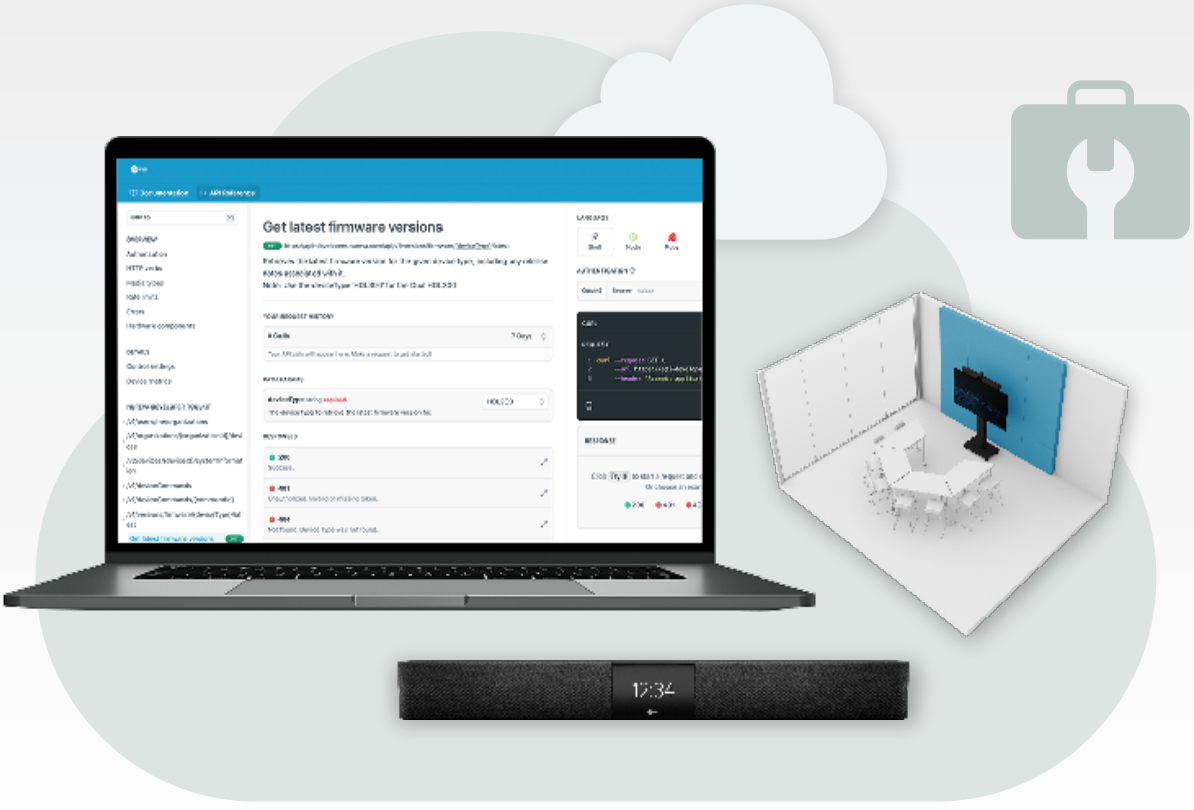
Nureva® WebSocket

LOCAL API GUIDE

This guide is a deprecated API version. For an updated version click here



CONTENTS

Local Documentation

Overview

Nureva® local APIs overview

Nureva's local APIs allow you to control your Nureva devices over your local network and access sound location data to share with camera integrations.

Local Control API

- Adjust the speaker volume of an audio device (Go to endpoint)
- Mute and unmute the microphone of an audio device (Go to endpoint)
- Adjust the treble and bass levels of an audio device (Go to endpoint)
- Recalibrate an audio device (Go to endpoint)

Sound location device API

- Get sound location data for an audio device, including location and power level (Go to endpoint)
- Get device information for an audio device, including the model and firmware version (Go to endpoint)
- Get layout information of the room where an audio device is located, including the dimensions of the room and the locations of the individual bars making up the system (Go to endpoint)

Note: Availability of features varies by device type and firmware version.

Local Control API Tutorial

Get started

How to connect

Clients need to make an initial WebSocket connection to the IP and port of the server. The IP can be found on Nureva® Console client. The port is 8932.

After a connection is established, clients can make requests to the server by using the request format described in the next section.

Notes

1. The WebSocket message is not compressed, so clients need to connect to the WebSocket server with the “perMessageDeflate:false” option.
2. For clients who wish to use a heartbeat on the WebSocket, the **ping** control frame is supported, per the WebSocket specification. **Ping** messages sent in accordance with the WebSocket specification will receive a **pong** response back. Calling other endpoints to achieve a heartbeat is not advised and may lead to adverse behaviour.
3. There is no automatic closing of this connection. It will be kept active as long as neither party has attempted to close it. Otherwise, we expect the connection to remain open and closure would be initiated by the integrating client.

Recalibrate device

This tutorial shows how to run recalibration on Nureva® HDL310 and HDL410 devices over the local network.

What does recalibration do?

Nureva audio devices have two types of calibration: automatic calibration and manual recalibration. In most cases, there is no need for users to perform any calibration procedures, since each audio device continuously auto-calibrates for optimal audio quality. Learn about automatic calibration.

Manual recalibration can be used to quickly optimize the audio performance of a device when the layout of the room changes significantly or the device is moved to a different room. It can also be included in a script to reset the room.

During the recalibration sequence, the speaker will generate a signal that sounds similar to static for approximately 20 seconds. Ensure the room is quiet when running recalibration.

It is possible but not recommended to recalibrate an audio device during a call. Call participants will not be able to hear each other while the recalibration is taking place. The microphones and speakers will be temporarily disconnected from the call. The call will resume after recalibration is complete.

For more information, see calibrating the HDL310 and calibrating the HDL410.

Overview

1. Use runRecalibration to make a request to run recalibration on a set of audio devices
2. Check that the request was successful

Sound Location Device API Tutorials

Get started

Authorization

Over the local network, no client authentication mechanisms are required. Requests are in plain text and no password or credentials are expected or supplied.

For requesting clients outside of the local host, the IP address or hostname is checked against an allowlist. Requests will be rejected if the IP address or hostname is not on the list.

How to configure the allowlist in Nureva® Console

See instructions under the tab labelled **'Setting up camera tracking - Nureva Console'** in this article:

Setting up camera tracking with Nureva® Console.

Start sound location data stream

This tutorial shows how to retrieve sound location data from Nureva® audio devices over the local network.

Sound location data returns information about the **loudest** sound detected in the room at a given time.

Once an initial request is made, sound location data registered by the audio device will start streaming at a regular interval.

What sound location data is available for HDL300, HDL310 and Dual HDL300 devices?

The sound location data provided for these devices consists of:

- **azimuth**: The audio direction angle, measured in degrees
- **powerLevel**: The sound power level, measured in decibels
- **time**: The timestamp when the location information was detected

Azimuth measurement

The audio direction angle is measured in degrees. The range is -70 to 70 degrees.

For HDL300 and HDL310 devices, this angle is relative to the center of the microphone and speaker bar.

For Dual HDL300 devices, this angle is relative to the center of the microphone and speaker bar that is connected to the port labeled "1" on the connect module.

Sound level

The sound level is measured in decibels (dB).

- Normal voice levels are typically above 40dB
- Sound levels below 40dB are typically background noises

Note: A sound level of 0 dB means that no meaningful sound has been detected in the room. Location data (azimuth) when sound level is 0 dB can be ignored.

What sound location data is available for HDL410 devices?

For HDL410 devices, sound location data consists of:

- **azimuth**: The audio direction angle, measured in degrees
- **powerLevel**: The sound power level, measured in decibels
- **coordinates**: The x, y coordinates of the location of the sound, in millimeters from the origin
- **time**: The timestamp when the location information was detected
- **triggeredZones**: The zone triggered by this sound location data event (if one exists). Returns an array that will contain either 0 or 1 elements. Contains the following fields:
    - **type**: The type of zone that was triggered. Currently, the only possible value of this string is 'Switching'
    - **label**: The label associated with the triggered zone
    - **id**: A unique identifier

⚠ Do not use azimuth data with HDL410

Although the **azimuth** field is provided for HDL410 devices, **it should not be used**. Use the **coordinates** field for HDL410 devices as it supports a wider variety of room configurations.

For forward compatibility, clients should use the **coordinates** field when available and fall back to use the **azimuth** field only when the **coordinates** field is unavailable.

Coordinate system

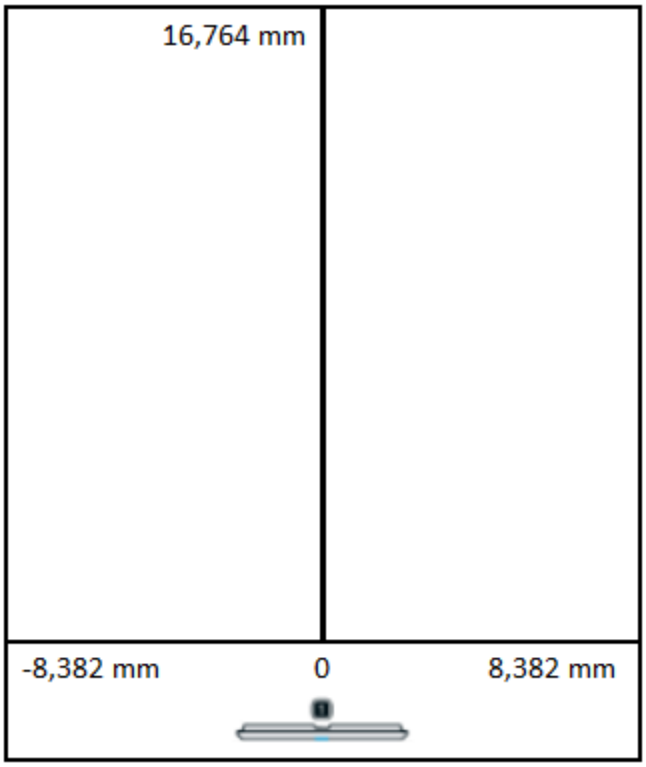
The location of the sound data is returned via x and y coordinates.

Position (0,0) represents the center of the microphone and speaker bar that is connected to the port labeled "1" on the connect module.

Actual maximum x and y values will vary by room, depending on the arrangement of the 2 microphone and speaker bars. However, the range of possible values is as follows:

- The x-coordinate values can be anywhere between -8,382 and 8,382 (8,382 mm to the left and to the right of the microphone and speaker bar plugged into port 1)
- The y-coordinate values can be anywhere between 1 and 16,764 (16,764 mm from the front of the microphone and speaker bar plugged into port 1)

The range of possible values is illustrated in the following diagram. Note that these values may not correlate to the size and shape of the actual room.



Sound level

The sound level is measured in decibels (dB).

- Normal voice levels are typically above 40dB
- Sound levels below 40dB are typically background noises

> Note: A sound level of 0 dB means that no meaningful sound has been detected in the room. Location data (azimuth and coordinates) when sound level is 0 dB can be ignored.

Triggered zones

Defining zones allows you to trigger events when sound is detected within a zone. For example, you can set up a meeting room system to select a particular camera angle when a particular zone is triggered.

If the detected sound is located within a defined zone, the **triggeredZones** field will return the information for that zone.

Triggered zone information includes:

- **type**: The zone type
    - There is currently only 1 zone type available: 'Switching'
    - Future releases will include other zone types
- **label:** The label given to the zone, defined by end users in Nureva Console
- **id**: A unique identifier

This data is returned as an array. When there are no elements in the array, it means that no zone was triggered. There can be at most 1 zone of a each type. Since there is currently only 1 defined zone type, the array will contain either 0 or 1 zones. This will change when more zone types are defined in future releases.

Learn more about zones

Get room layout information

Only HDL410 systems support this feature.

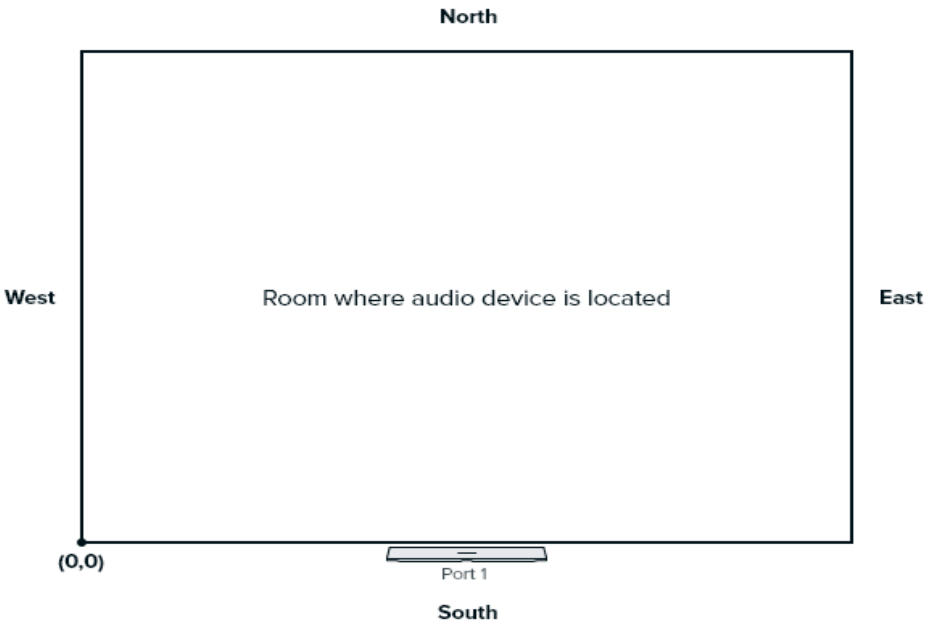
This tutorial shows how to get room layout information about the room where Nureva® audio devices are located.

What room layout information is available?

As shown in the image below, rooms are assumed to be rectangular, and each wall is identified by a cardinal direction: north, south, west and east.

When interpreting the microphone and speaker bar location coordinates, the origin (0,0) is the southwest corner of the room. Positive x values are in the east direction. Positive y values are in the north direction.

The microphone and speaker bar plugged into Port 1 on the connect module will always be situated on the south wall of the room layout.



Note that room dimensions are configurable by end users through Nureva Console software, whereas bar locations are detected through the calibration process during hardware installation. As a result, the bar locations are not guaranteed to align precisely with a wall.

roomDimensions	The dimensions of the room in millimeters, returned as x and y values. • x value: west-east dimension (horizontal) • y value: south-north dimension (vertical)
bars	An ordered array: • The first element is the microphone and speaker bar connected to Port 1 on the connect module • The second element is the microphone and speaker bar connected to Port 2 on the connect module
wall	The wall on which each microphone and speaker bar is located. Possible values: • North • South • East • West Note: Not to be confused with the direction the bar is facing. For example: • If the bar is located on the south wall, then it is facing north • If the bar is located on the west wall, then it is facing east
location	The x, y coordinates of each microphone and speaker bar's location, measured in millimeters • The origin (0,0) of the coordinate system is the southwest corner of the room • Positive x values are in the east direction • Positive y values are in the north direction

Room layout coordinates vs. sound location data coordinates

The room layout's origin differs from the origin used in the sound location data stream

The origin for the sound location data stream is the location of the Port 1 microphone and speaker bar, while the origin for the room layout is the southwest corner of the room.

To convert a sound location coordinate (x, y) to a room coordinate, use (x + x', y), where x' is the x value of the Port 1 bar's location in the response's bars array.

Overview

Use Get room layout information to get the room layout information.

Get room zones information

> Only HDL410 systems support this feature.

This tutorial shows how to get zone information for the room where Nureva® audio devices are located.

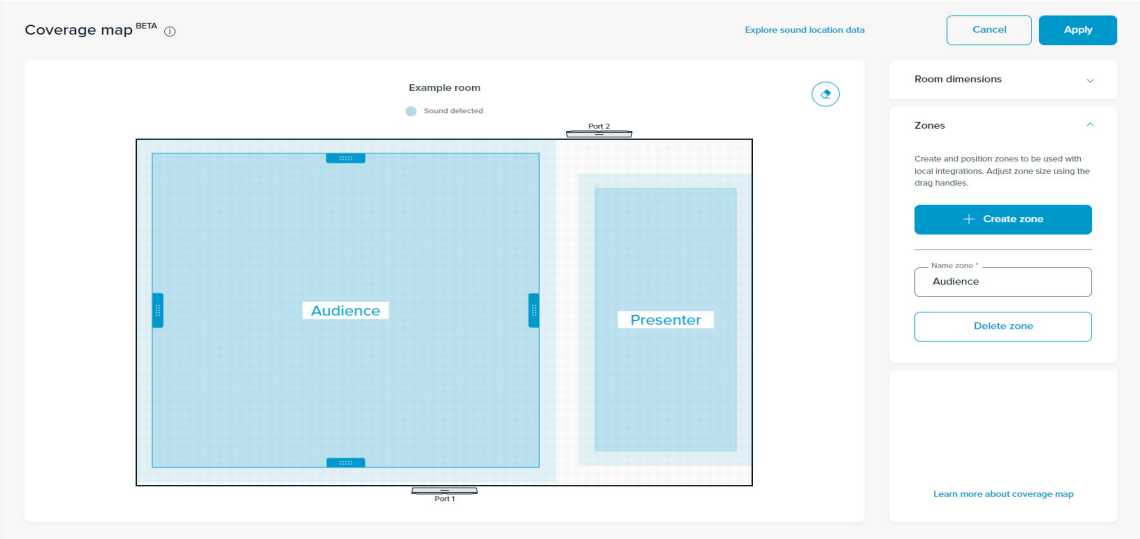
What is a zone?

A zone is an area within a room. Defining zones allows you to trigger events when sound is detected within a zone. For example, you can set up a meeting room system to select a particular camera angle when a particular zone is triggered.

How to configure zones in Nureva Console

The sound location data stream endpoint will return zone data when any zones are triggered. Learn more about triggered zones.

In the example room below, the presenter stands at the front of the room while the audience comprises the remainder of the room. Employing zones in this room allows one camera to be focused on the presenter while a second camera points at the audience. During a meeting, the appropriate camera view can be selected based on whether sound is currently detected in the presenter zone versus in the audience zone.



Number of zones

A maximum of 8 zones can be created in each room.

Overlapping zones

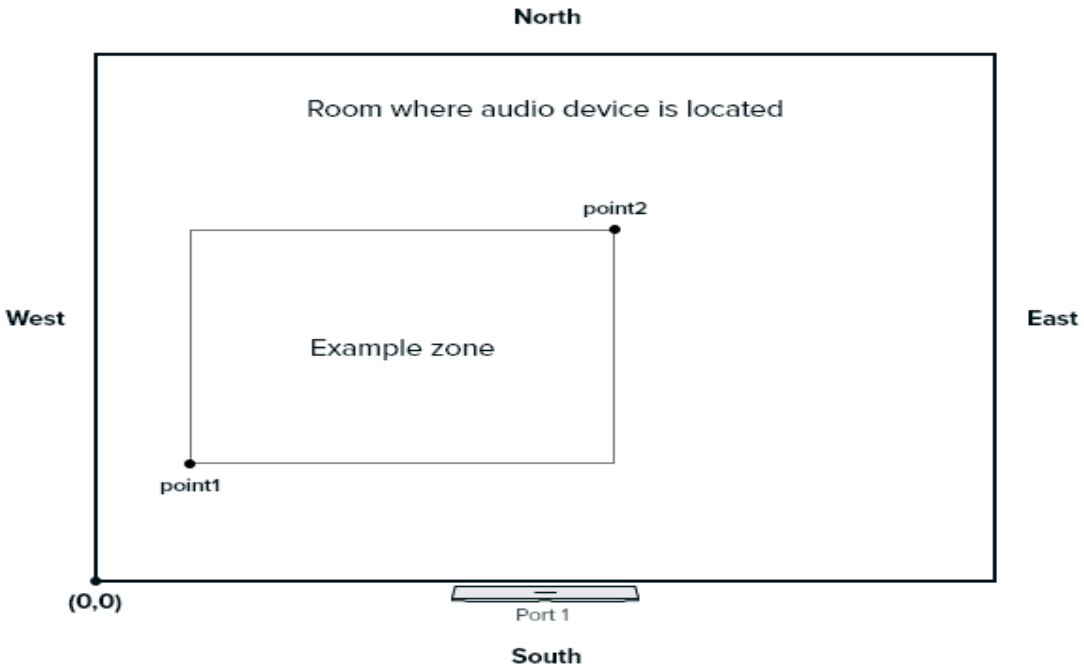
Zones of the same type cannot overlap.

Zone position

All zones are rectangular. Zone position is defined by 2 points:

- **point1** gives the (x,y) coordinate of one corner of the zone's bounding box, in millimeters
- **point2** gives the (x,y) coordinate of the opposite corner of the zone's bounding box, in millimeters

The same coordinate system is used for both zone position and room layout information. As shown in the image below, rooms are assumed to be rectangular, and each wall is identified by a cardinal direction: north, south, west and east.



When interpreting the zone position coordinates, the origin (0,0) is the southwest corner of the room. Positive x values are in the east direction. Positive y values are in the north direction.

The microphone and speaker bar plugged into Port 1 on the connect module will always be situated on the south wall of the room layout.

What zone information is available?

Each zone has a type, a position (geometry), a label and a unique id.

types	An array of 1 or more strings defining the purpose of this zone. Presently, the only possible type of zone is 'Switching'. This indicates that this zone is intended to define support for camera switching within this room. It is not possible for the same type of zone to appear more than once within this array.
geometry	Position of the zone defined by a set of coordinates where the southwest corner of the room is taken as the origin (0,0) of the coordinate system. All zones are rectangular or square with their bounding box defined by the **point1** and **point2** properties on opposite corners. • **point1**: The **x** and **y** location, in room coordinates, of one corner of the zones' bounding box in in millimeters. **point1** is the opposite corner to **point2.** • **point2**: The **x** and **y** location, in room coordinates, of one corner of the zones' bounding box in in millimeters. **point2** is the opposite corner to **point1.**
label	The label given to the zone, defined by end users in Nureva Console
id	A unique identifier

Overview

Use Get room zones information to get the room layout information.

Integrations

Camera tracking integrations

Nureva's sound location device API provides accurate data collected by Nureva® audio systems to be shared with third-party cameras.

Nureva has collaborated with leading camera companies to integrate the use of sound location data with supported cameras. This requires configuring Nureva Console and setting up the third-party camera's software.

The following manufacturers use Nureva sound location data to drive camera views:

How to set up camera tracking in Nureva Console

Visit Setting up camera tracking with Nureva Console.

AVer camera tracking

AVer software allows supported AVer cameras to steer their view based on the microphone pickup in the room.

For more information, visit Integrating AVer cameras with camera tracking.

Crestron sound tracking module

A Crestron sound tracking module is available to support using sound location data from the HDL300, Dual HDL300, HDL310 and HDL410 with selected cameras and Crestron systems. Sound location data can be used by cameras that have a Crestron module and support "preset recalls."

For more information, visit Using the Crestron® sound tracking module with camera tracking.

Extron sound tracking driver

Extron has created a sound tracking driver to support using sound location data from the HDL300, Dual HDL300 and HDL410 with selected cameras and Extron systems. Sound location data can be used by cameras that have an Extron driver in place and support "preset recalls."

For more information, visit Using the Extron® driver with camera tracking.

Lumens® camera control

CamConnect software synchronizes Lumens PTZ camera control with sound location data from HDL300 and Dual HDL300 systems. This allows Lumens PTZ cameras to switch focus to the active-speaking participant in a meeting space.

For more information, visit Integrating Lumens® PTZ cameras with camera tracking.

Q-SYS™ camera control plugin

Nureva has developed a Q-SYS-certified plugin for its HDL310 and HDL410 audio-conferencing systems to deliver audio control and camera integration functionality that will greatly enhance and simplify the remote experience. The Q-SYS plugin enables customers to adjust the Nureva® HDL pro series systems from the Q-SYS audio, video and control platform.

Sound location data from the HDL310 and HDL410 systems is included in the plugin and can be used to automate Q-SYS cameras without additional programming.

For more information, go to Using the Q-SYS control plugin.

Local control system integrations

Nureva®'s audio device control API allows third-party control systems to adjust audio settings such as speaker volume and microphone mute.

The following manufacturers have developed integrations that allow control of Nureva audio devices via the local area network:

Crestron® control module

Nureva® has developed and released a control module for the HDL310 and HDL410 audio conferencing systems. This is a Crestron 3 and 4-series SIMPL# module and is available from the Crestron Application Marketplace.

What functions are available through the module?

The control module allows users to control Nureva devices that are connected to a Crestron series 3 or 4 controller. The following functions are available:

Function	HDL310	HDL410
Microphone mute/unmute	✓	✓
Volume up/down	✓	✓
Recalibration	✓	✓
Adjust treble/bass	✓	✓

Before downloading the module

Update the Nureva audio conferencing system to the latest firmware version

- How to update HDL310 and HDL410 firmware

Download the control module

Download and install the Crestron local control module on your Crestron 3 and 4-series SIMPL# module.

- Download the Crestron control module

Extron control system driver

Extron provides local control system drivers for HDL310 and HDL410 audio conferencing systems.

Access to these drivers requires registration in the Extron Insider program.

What functions are available through the driver?

The control system driver allows users to control Nureva® devices directly from Extron TouchLink® Pro Touchpanels that are connected to Pro Series controllers. The following functions are available:

Function	HDL310	HDL410
Microphone mute/unmute	✓	✓
Volume up/down	✓	✓
Recalibration	✓	✓
Adjust treble/bass	✓	✓

Before downloading the driver

Update the Nureva audio conferencing system to the latest firmware version

- How to update HDL310 and HDL410 firmware

Download the control system driver

Download and install the appropriate Extron control system driver on your Pro Series enabled product by following these steps:

1. Become an Extron insider
2. Go to Extron Control System Drivers
3. Select your IP Link® Pro Series product from the **Extron Product** dropdown
4. In the **Manufacturer** dropdown, select **Nureva**
5. Download the appropriate driver for your Nureva device:
    i. HDL310: Download the driver named **Nureva - HDL310**
    ii. HDL410: Download the driver named **Nureva - HDL410**
    iii. Do **not** use the 'Sound Tracking' drivers
6. Use the Extron tools to connect controller actions to Nureva devices
    i. Completion of Extron's Control Specialist training is recommended

Q-SYS™ control plugin

Nureva has developed a Q-SYS-certified local control plugin for the HDL310 and HDL410 audio conferencing systems. This plugin is accessed through the Q-SYS™ Designer Asset Manager.

What functions are available through the plugin?

The Q-SYS plugin allows users to monitor and manage Nureva® devices with the Q-SYS Reflect Enterprise Manager. The following functions are available:

Function	HDL310	HDL410
Microphone mute/unmute	✓	✓
Volume up/down	✓	✓
Recalibration	✓	✓
Adjust treble/bass	✓	✓

Before downloading the plugin

Update the Nureva audio conferencing system to the latest firmware version.

- How to update HDL310 and HDL410 firmware

Download the control plugin

Download and install the Q-SYS Designer Asset Manager by following these steps:

- Create a QSC account
- Go to the Q-SYS Nureva technology partner site
- Scroll down to the Nureva HDL310 and HDL410 section and click the Q-SYS Designer Asset Manager download link
- On the Q-SYS Designer Software page, scroll down toward the bottom and download the latest version of Q-SYS Designer Software

Technical Details

Connection limits

The server itself has an internal limit on the number of open connections that can be made at any one time.

Currently, this limit is **5 open connections at a time**.

Additional connection attempts will be rejected until an open connection becomes available.

Local API Reference

Overview

Errors

Error responses sent to client requests follow this format:

JSON

```
{
      “request”: string,
       “requestId”: string,
      “clientId”: string,
      “errors”: {
      “statusCode”: number,
      “message”: string
 }[ ]
 }
```

request/requestId/clientId: The value is echoed back from the original request made to the service unless it was invalid.

statusCode: A numeric code indicating the specific error encountered with the request.

message: Contextual information about the error and what prompted it.

Local Control API errors

Some errors that may occur when developing with the Nureva® local control API:

Code	Description
3000	Socket closed: Unauthorized access
4050	Socket closed: Maximum number of connected clients reached
4051	Socket closed: Requesting IP is not in valid range
10000	Bad request (e.g., invalid request format, invalid parameters)
10001	Unexpected error preventing request fulfillment
12000	Speaker bar is disconnected
12001	Unsupported attributes

Sound location device API errors

Some errors that may occur when developing with the Nureva® sound location device API:

Code	Description
3000	Socket closed: Unauthorized access
4050	Socket closed: Maximum number of connected clients reached
4051	Socket closed: Requesting IP is not in valid range
10000	Bad request (e.g., invalid request format, invalid parameters)
10001	Unexpected error preventing request fulfillment
11000	Unsupported device
11001	Device not connected
11002	Interrupted by firmware update
11003	Audio location unavailable
11004	Microphone is muted
11005	Speaker bar is disconnected
11006	Room info unavailable

Local Control API

v1/devices/commands/getControlSettings

Get control settings

> The Nureva® sound location device API is for use with HDL300, HDL310, Dual HDL300 and HDL410 devices.

Allows clients to get control settings from a device, including the speaker treble level, the speaker bass level, and whether or not the microphone is muted.

After a request is made, the control settings will be returned via the response body from the device based on which settings were requested. If there is no specified body, or no attributes in the attributes array inside the request body, then it will return all control settings supported.

If an error occurs, the WebSocket will respond with an appropriate status code and message.

Control Setting	Description
microphoneMute	When this property is true, the microphone is muted. When this property is false, the microphone is not muted.
speakerTrebleLevel	This is the speaker treble level. This is measured in whole numbers from 0 to 100.
speakerBassLevel	This is the speaker bass level. This is measured in whole numbers from 0 to 100.

Request

JSON

```
{
    “request”: “v1/devices/commands/getControlSettings” ,
    “requestId”: string,
    “clientId”: string,
    “body”: {
        “attriutes”: [“microphoneMute”, “speakerTrebleLevel” , “speakerBassLevel”]
    }
}
```

Response

JSON

```
{
    “request”: “v1/devices/commands/getControlSettings” ,
    “requestId”: string,
    “clientId”: string,
    “body”: {
        “microphoneMute”: boolean,
        “speakerTrebleLevel”: number,
        “speakerBassLevel”: number,
    }
}
```

Errors

Status code	Description
10000 – Bad request	Request does not conform to the specification. The message field will include detailed information about which part of the request is incorrect.
12001 – Unsupported attributes	Request body does not conform to the specification. The message field will include detailed information about which part of the request body is incorrect.

Sample requests

1. Valid request/response with body

Request

JSON

```
{
   "request": "v1/devices/commands/getControlSettings",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": "test",
   "body": {
      "attributes": ["microphoneMute"]
   }
}
```

Response

JSON


```
{
    "request": "v1/devices/commands/getControlSettings",
    "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
    "clientId": "test",
    "body": {
        "microphoneMute": true
    }
}
```


2. Valid request/response with no body

Request

JSON


```
{
    "request": "v1/devices/commands/getControlSettings",
    "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
    "clientId": "test"
}
```

Response

JSON


```
{
   “request”: “v1/devices/commands/getControlSettings”,
   “requestId”: “3abe203s-42b7-4b0b-9awaf-5c381793a192”,
   “clientId”: “test”,
   “body”: {
      ”microphoneMute”:  false,
      “speakerTrebleLevel”: 48,
      “speakerBassLevel”: 55
   }
}
```


3. Invalid request - no requestId or clientId

Request

JSON


```
{
   “request”: “v1/devices/commands/getControlSettings”,
   “requestId”: “”,
   “clientId”: “”
}
```

Response

JSON


```
{
  "request": "v1/devices/commands/getControlSettings",
  "requestId": " ",
  "clientId": " ",
  "errors": [
  {
   "statusCode": 10000,
   "message": "clientId: String can't be empty or whitespace"
  },
  {
   "statusCode": 10000,
   "message": "requestId: String can't be empty or whitespace"
  }
 ]
}
```


4. Invalid request - bad body

This is an invalid request, since the attributes values are all case sensitive.

Request

JSON


```
{
  "request": "v1/devices/commands/getControlSettings",
  "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
  "clientId": "test",
  "body": {
     "attributes": ["microphonemute"]
  }
}
```

Response

JSON


```
{
  “request”: “v1/devices/commands/getControlSettings”,
  “requestId”: ”3abe203s-42b7-4b0b-9awaf-5c381793a192”,
  “clientId”: “test”,
  “errors”: [
  {
    “statusCode”: 12001,
    “message”: “Unsupported Attributes: attributes 0. Invalid enum value. Expected ‘microphoneMute’ | ‘speakerTrebleLevel’ | ‘speakerBassLevel’, received ‘microphonemute’”
  }
 ]
}
```

v1/devices/commands/setControlSettings

Set control settings

Only HDL310 and HDL410 devices support this command.

Allows clients to set device control settings, including the speaker treble level, the speaker bass level, and whether or not the microphone is muted.

A request body must be included with the control settings and values. If the request was successful, the request, requestId and clientId will be echoed back as a response.

If there is no request body, incorrect control settings, or an error has occurred, then the WebSocket will respond with an appropriate status code and message.

Control Setting	Description
microphoneMute	When this property is true, the microphone is muted. When this property is false, the microphone is not muted.
speakerTrebleLevel	This is the speaker treble level. This is measured in whole numbers from 0 to 100.
speakerBassLevel	This is the speaker bass level. This is measured in whole numbers from 0 to 100.

Request

JSON

```
{
   “request”: “v1/devices/commands/setControlSettings” ,
   “requestId”: string,
   “clientId”: string,
   “body”: {
      “microphoneMute”: boolean,
      “speakerTrebleLevel”: number,
      “speakerBassLevel”: number
   }
}
```

Response

JSON

```
{
   “request”: “v1/devices/commands/setControlSettings” ,
   “requestId”: string,
   “clientId”: string,
}
```

Errors

Status code	Description
10000 – Bad request	Request does not conform to the specification. The message field will include detailed information about which part of the request is incorrect.
12001 – Unsupported attributes	Request body does not conform to the specification. The message field will include detailed information about which part of the request body is incorrect.

Sample requests

1. Valid request/response with one setting in body.

Sets the microphone to unmuted. The response indicates that the change was successfully applied.

Request

JSON

```
{
   "request": "v1/devices/commands/setControlSettings",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": "test",
   "body": {
      "microphoneMute":  false
   }
}
```

Response

JSON

```
{
    "request": "v1/devices/commands/setControlSettings",
    "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
    "clientId": "test"
}
```

2. Valid request/response with all settings in body

Sets the microphone to unmuted, the treble level to 80, and the bass level to 30. The response indicates that the change was successfully applied.

Request

JSON

```
{
    "request": "v1/devices/commands/setControlSettings",
    "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
    "clientId": "test",
    "body": {
        "microphoneMute":  false,
        "speakerTrebleLevel": 80,
        "speakerBassLevel": 30
    }
}
```

Response

JSON


```
{
  "request": "v1/devices/commands/setControlSettings",
  "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
  "clientId": "test"
}
```


3. Invalid request - no requestId or clientId

Request

JSON


```
{
  "request": "v1/devices/commands/setControlSettings",
  "requestId": "",
  "clientId": ""
}
```

Response

JSON


```
{
  "request": "v1/devices/commands/setControlSettings",
  "requestId": " ",
  "clientId": " ",
   "errors": [
  {
   "statusCode": 10000,
   "message": "clientId: String can't be empty or whitespace"
  },
  {
   "statusCode": 10000,
   "message": "requestId: String can't be empty or whitespace"
  }
 ]
}
```


4. Invalid request - bad body

This is an invalid request, since the control settings expect specific value types and ranges.

Request

JSON


```
{
   "request": "v1/devices/commands/setControlSettings",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": "test",
   "body": {
       "microphoneMute":  "a",
       "speakerTrebleLevel": -1.2,
       "speakerBassLevel": 101
   }
}
```

Response

JSON


```
{
  "request": "v1/devices/commands/setControlSettings",
  "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
  "clientId": "test",
  "errors": [
  {
    "statusCode": 12001,
    "message": "Unsupported Attributes: microphoneMute. Expected boolean, received string"
  },
  {
    "statusCode": 12001,
    "message": "Unsupported Attributes: speakerTrebleLevel. Error, speakerTrebleLevel  must be a whole number"
  },
  {
    "statusCode": 12001,
    "message": "Unsupported Attributes: speakerTrebleLevel. Error, speakerTrebleLevel cannot be less than 0"
  },
  {
    "statusCode": 12001,
    "message": "Unsupported Attributes: speakerBassLevel. Error, speakerBassLevel cannot be greater than 100"
  }
 ]
}
```

v1/devices/commands/changeSpeakerVolume

Change speaker volume

Only HDL310 and HDL410 devices support this command.

Allows clients to increment or decrement volume on a device by 4-6 every time a command is sent. The lowest volume is 0 and the highest is 100.

A request body must be included. The only supported operations are 'increment' and 'decrement' (case sensitive). If the request was successful, the request, requestId and clientId will be echoed back as a response.

If there is no valid request body, or an error has occurred, then the WebSocket will respond with an appropriate status code and message.

Operation	Description
increment	This will attempt to increment the speaker volume by 4-6, to a maximum of 100.
decrement	This will attempt to decrement the speaker volume by 4-6, to a minimum of 0.

Request

JSON

```
{
   “request”: “v1/devices/commands/changeSpeakerVolume”,
   “requestId”: string,
   “clientId”: string,
   “body”: {
      “operation”: string
   }
}
```

Response

JSON

```
{
   “request”: “v1/devices/commands/changeSpeakerVolume”,
   “requestId”: string,
   “clientId”: string,
}
```

Errors

Status code	Description
10000 – Bad request	Request does not conform to the specification. The message field will include detailed information about which part of the request is incorrect.
12001 – Unsupported attributes	Request body does not conform to the specification. The message field will include detailed information about which part of the request body is incorrect.

Sample requests

1. Valid request/response to increment the speaker volume

Request

JSON

```
{
    "request": "v1/devices/commands/changeSpeakerVolume",
    "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
    "clientId": "test",
    "body": {
        "operation": "increment",
    }
}
```

Response

JSON


```
{
    “request”: “v1/devices/commands/changeSpeakerVolume”,
    “requestId”: “3abe203s-42b7-4b0b-9awaf-5c381793a192”,
    “clientId”: “test”
}
```


2. Valid request/response to decrement the speaker volume

Request

JSON


```
{
    “request”: “v1/devices/commands/changeSpeakerVolume”,
    “requestId”: “3abe203s-42b7-4b0b-9awaf-5c381793a192”,
    “clientId”: “test”,
    “body”: {
        “operation”: “decrement”,
    }
}
```

Response

JSON


```
{
  "request": "v1/devices/commands/changeSpeakerVolume",
  "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
  "clientId": "test"
}
```


3. Invalid request - no requestId or clientId

Request

JSON


```
{
  "request": "v1/devices/commands/changeSpeakerVolume",
  "requestId": "",
  "clientId": ""
}
```

Response

JSON


```
{
   "request": "v1/devices/commands/changeSpeakerVolume",
   "requestId": " ",
   "clientId": " ",
    "errors": [
   {
    "statusCode": 10000,
    "message": "clientId: String can't be empty or whitespace"
   },
   {
    "statusCode": 10000,
    "message": "requestId: String can't be empty or whitespace"
   }
 ]
}
```


4. Invalid request - bad body

This is an invalid request, since the only operations supported are increment and decrement.

Request

JSON


```
{
   "request": "v1/devices/commands/changeSpeakerVolume",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": "test",
   "body": {
       "operation":  "min",
   }
}
```

Response

JSON


```
{
  “request”: “v1/devices/commands/changeSpeakerVolume”,
  “requestId”: “3abe203s-42b7-4b0b-9awaf-5c381793a192”,
  “clientId”: “test”,
  “errors”: [
  {
    “statusCode”: 12001,
    “message”: “Unsupported Attributes: operation. Invalid enum value. Expected ‘increment’ | ‘decrement’, received ‘min’”
  }
 ]
}
```

v1/devices/commands/runRecalibration

Run recalibration

Only HDL310 and HDL410 devices support this command.

Runs the recalibration for the connected Nureva® audio device. The device will emit a loud static-like sound for approximately 20 seconds. Ensure the room is quiet when running recalibration. We do not recommend recalibrating when the device is in a call. The speaker bars must be connected to the CM2 for the recalibration to start.

JSON

```
{
  "request": "v1/devices/commands/runRecalibration",
  "requestId": string,
  "clientId": string,
}
```

The response below is returned on a successful request to run recalibration. The content of the response is an echo of the original response.

Response

JSON

```
{
  "request": "v1/devices/commands/runRecalibration",
  "requestId": string,
  "clientId": string,
}
```

Errors

Status code	Description
10000 – Bad request	Request does not conform to the specification. The message field will include detailed information about which part of the request is incorrect.
12000 – Speaker bar disconnected	Request body does not conform to the specification. The message field will include detailed information about which part of the request body is incorrect.

Sample requests

1. Valid request/response

Request

JSON


```
{
  "request": "v1/devices/commands/runRecalibration",
  "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a195",
  "clientId": "Extron"
}
```


Response

JSON


```
{
  "request": "v1/devices/commands/runRecalibration",
  "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a195",
  "clientId": "Extron"
}
```

2. Bad request - no requestId or clientId

Request

JSON

```
{
  "request": "v1/devices/commands/runRecalibration",
  "requestId": "",
  "clientId": ""
}
```

Response

JSON

```
{
  "request": "v1/devices/commands/runRecalibration",
  "requestId": " ",
  "clientId": " ",
   "errors": [
  {
   "statusCode": 10000,
   "message": "clientId: String can't be empty or whitespace"
  },
  {
   "statusCode": 10000,
   "message": "requestId: String can't be empty or whitespace"
  }
 ]
}
```

3. Speaker bar disconnected

This is a valid request, but returns an error because the speaker bars have been disconnected.

Request

JSON

```
{
   "request": "v1/devices/commands/runRecalibration",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a195",
   "clientId": "Extron"
}
```

Response

JSON

```
{
   "request": "v1/devices/commands/runRecalibration",
   "requestId": " ",
   "clientId": " ",
    "errors": [
  {
    "statusCode": 12000,
    "message": "Speaker bar disconnected"
  }
 ]
}
```

Sound Location Device API

v1/devices/audioLocation

Get sound location data stream

Allows clients to retrieve sound location data, including audio direction angle, sound power level and (in some cases) the coordinates of the location.

Once the initial request is made, sound location data registered by the connected audio device will start streaming at a regular interval. The data will be streamed provided the WebSocket connection is open and the data can be retrieved.

If an error occurs, the WebSocket will respond with the appropriate status code. If a Nureva® device firmware update is in progress, a status code will be returned and sound location data streaming will be paused. Once the firmware update is complete, sound location data streaming will resume automatically.

Request

JSON

```
{
    "request": "v1/devices/audioLocation",
    "requestId": string,
    "clientId": string,
}
```

Response

JSON

```
{
   “request”: “v1/devices/audioLocation”,
   “requestId”: string,
   “clientId”: string,
   “body”: {
      “azimuth”: number,
      “powerLevel”: number,
      “coordinates”: { “x”: number, “y”: number } | undefined,
      “time”: string,
      “triggeredZones”: [ {
            “type”: “Switching”,
             “label”: string,
             “id”: string,
        } [ ] | undefined
   }
}
```

Use the azimuth field **only** for HDL300, HDL310 and Dual HDL300 systems.

Use the coordinates field **instead** of the azimuth field for HDL410 systems.

azimuth	*Only for use with HDL300, HDL310 and Dual HDL300 systems.* Audio direction angle in degrees. Range -70 to 70, e.g., 65.
powerLevel	Sound power level in decibels, e.g., 30.
coordinates	Available for HDL410 systems only. The **x**, **y** coordinates of the location in millimeters. Bar 0's (i.e., Port 1) location is taken as the origin (0, 0) of the coordinate system, where positive **y** values are positioned in front of the bar. Negative and positive **x** values are to the left and right of the bar, respectively, when facing in the same direction as the bar.
time	ISO 8601 time string for the time the location information was generated, e.g., "2022-06-07T09:01:23.486Z".
triggeredZones	*Available for HDL410 systems only.* An array of 0 or more zones that have been triggered by this sound location data event. The array will only ever have elements for HDL410 devices that have zones configured by an end user through Nureva Console and there will be at most 1 zone of a given type. • **type**: The type of zone that was triggered. The only possible value of this string, at this time, is the value of "Switching". • **label**: The label associated with the triggered zone. • **id**: A unique identifier for the triggered zone that is not intended to be human readable.

Errors

Status code	Description
10000 – Bad request	Request does not conform to the specification. The message field will include detailed information about which part of the request is incorrect.
11000 – Unsupported device	Connected hardware device does not support audio location streaming.
11001 – Device not connected	No audio device is connected.
11002 – Interrupted by firmware update	If a firmware update is being applied to a connected Nureva device, audio location information will be unavailable until that update is complete. An error code will be supplied.
11003 – Audio location unavailable	Audio location information is not available. A general use error in the event we cannot retrieve and send audio location.
11004 – Microphone is muted	Microphone is muted.
11005 – Speaker bar is disconnected	Speaker bar is disconnected.

Sample requests

1. Valid request/response

Request

JSON

```
{
  “request”: “v1/devices/audioLocation”,
  “requestId”: “3abe203s-42b7-4b0b-9awaf-5c381793a192”,
  “clientId”: “test”,
}
```

Response

JSON


```
{
    “request”: “v1/devices/audioLocation”,
    “requestId”: “3abe203s-42b7-4b0b-9awaf-5c381793a192”,
    “clientId”: “test”,
    “body”: {
        “azimuth”:  -35,
        ”powerLevel”:  0,
        “coordinates”: {
                “x”: -2890,
                “y”: 2936
        },
        “time”: “2022-07-15T15:07:23.375Z”,
        “triggeredZones”: [{
                “type”: “Switching”,
                “label”: “Zone 1”,
                “id”: “unique identifier”
        } ]
    }
}
```

2. Invalid request - no requestId or clientId

Request

JSON


```
{
   “request”: “v1/devices/audioLocation”,
   “requestId”: “”,
   “clientId”: “”
}
```


Response

JSON


```
{
   “request”: “v1/devices/audioLocation”,
   “requestId”: “ ”,
   “clientId”: “ ”,
    “errors”: [
      {
            “statusCode”: 10000,
            “message”: “clientId: String can’t be empty or whitespace”
       },
      {
            “statusCode”: 10000,
            “message”: “requestId: String can’t be empty or whitespace”
      }
   ]
}
```

3. Unsupported device

This is a valid request, but returns an error because an unsupported audio device is connected.

Request

JSON

```
{
    "request": "v1/devices/audioLocation",
    "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
    "clientId": "test"
}
```

Response

JSON

```
{
    "request": "v1/devices/audioLocation",
    "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
    "clientId": "test",
     "errors": [
       {
            "statusCode": 11000,
            "message": "Unsupported device"
       }
   ]
}
```

v1/devices/info

Get device information

Returns device information for the connected Nureva® audio conferencing system, including the model and firmware version.

Request

JSON

```
{
    "request": "v1/devices/info",
    "requestId": string,
    "clientId": string,
}
```

Response

JSON

```
{
    "request": "v1/devices/info",
    "requestId": string,
    "clientId": string,
    "body": {
        "model": string,
        "firmwareVersion": string,
    }
}
```

model	The model of the default device. Possible values: • hdl300 • dual-hdl • hdl310 • hdl410
firmwareVersion	The firmware version of the default device. Format: x.y.z

Errors

Status code	Description
10000 – Bad request	Request does not conform to the specification. The message field will include detailed information about which part of the request is incorrect.
11000 – Unsupported device	Connected hardware device does not support audio location streaming.
11001 – Device not connected	No audio device is connected.

Sample requests

1. Valid request/response

Request

JSON


```
{
    “request”: “v1/devices/info”,
    “requestId”: “3abe203s-42b7-4b0b-9awaf-5c381793a192”,
    “clientId”: “test”,
}
```


Response

JSON


```
{
    “request”: “v1/devices/info”,
    “requestId”: “3abe203s-42b7-4b0b-9awaf-5c381793a192”,
    “clientId”: “test”,
    “body”: {
        “firmwareVersion” : “3.1.9”,
        “model”: “hdl300”,
    }
}
```

2. Invalid request - no requestId or clientId

Request

JSON


```
{
   “request”: “v1/devices/info”,
   “requestId”: “”,
   “clientId”: “”
}
```


Response

JSON


```
{
   “request”: “v1/devices/info”,
   “requestId”: “ ”,
   “clientId”: “ ”,
    “errors”: [
      {
            “statusCode”: 10000,
            “message”: “clientId: String can’t be empty or whitespace”
      },
      {
            “statusCode”: 10000,
            “message”: “requestId: String can’t be empty or whitespace”
      }
   ]
}
```

3. clientId parameter is too long

Request

JSON

```
{
   “request”: “v1/devices/info”,
   “requestId”: “”,
   “clientId”: “This_id_is_longer_than_fifty_characters_and_is_too_long!“
}
```

Response

JSON

```
{
   “request”: “v1/devices/info”,
   “requestId”: “ ”,
   “clientId”: “ ”,
    “errors”: [
      {
            “statusCode”: 10000,
            “message”: “clientId: Storing must contain at most 50 character(s)”
      },
      {
            “statusCode”: 10000,
            “message”: “requestId: String can’t be empty or whitespace”      }
  ]
}
```

4. No device connected

This is a valid request, but returns an error because no audio device is connected.

Request

JSON

```
{
   "request": "v1/devices/info",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": "test"
}
```

Response

JSON

```
{
   "request": "v1/devices/info",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": "test ",
    "errors": [
      {
            "statusCode": 11001,
            "message": "Device not connected"
      }
   ]
}
```

5. Unsupported device

This is a valid request, but returns an error because an unsupported audio device is connected.

Request

JSON

```
{
   "request": "v1/devices/info",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": "test"
}
```

Response

JSON

```
{
   "request": "v1/devices/info",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": "test",
    "errors": [
      {
            "statusCode": 11000,
            "message": "Unsupported device"
      }
  ]
}
```

v1/room/layout

Get room layout information

Only HDL410 systems support this feature.

Returns layout information of the room where the connected Nureva® audio conferencing system is located. The information includes dimensions of the room and locations of the individual microphone and speaker bars making up the system.

Rooms are assumed to be rectangular, and each wall is identified by a cardinal direction: north, south, west and east for the top, bottom, left and right wall, respectively. When interpreting the bar location coordinates, the southwest corner of the room is taken as the origin (0, 0).

⚠ The room layout's origin differs from the one in the sound location data stream, which uses Bar 0's location as the origin.

Bar 0 (i.e., the microphone and speaker bar plugged into Port 1 in the connect module) will always be situated on the south wall of the room layout. Therefore, to convert a sound location coordinate (x, y) to a room coordinate, use (x + x', y), where x' is the x value of Bar 0's location in the response's bars array. Each index in the bars array corresponds to that bar's order in the system, so Bar 0 is the first element and Bar 1 is the second element.

Note that room dimensions are configurable by end users through Nureva Console software, whereas bar locations are detected through the calibration process during hardware installation. As a result, the bar locations are not guaranteed to align precisely with a wall.

Request

JSON

```
{
    "request": "v1/room/layout",
    "requestId": string,
    "clientId": string,
}
```

Response

JSON

```
{
    "request": "v1/room/layout",
    "requestId": string,
    "clientId": string,
    "body": {
        "roomDimensions": {
            "x": number,
            "y": number,
        },
        "bars": {
            "wall": "North" | "South" | "West" | "East",
            "location": {
                "x": number,
                "y": number,
        }
    } [ ]
  }
}
```

roomDimensions	The dimensions of the room in millimeters. The x value represents the west-east dimension and the y value represents the north-south dimension.
bars	An ordered array where each element is the location for a single bar in the system. Each index corresponds to that bar's order in the system (e.g., first element is Bar 0, second element is Bar 1).
wall	The wall the bar is located on. Not to be confused with the direction the bar is facing. For example, if it is located on the south wall, then it is facing north; if it is on the west wall, then it is facing east.
location	The **x**, **y** coordinates of the bar's location in millimeters. The southwest corner of the room is taken as the origin (0, 0) of the coordinate system, where positive **y** values are in the north direction and positive **x** values are in the east direction.

Errors

Status code	Description
10000 – Bad request	Request does not conform to the specification. The message field will include detailed information about which part of the request is incorrect.
11000 – Unsupported device	Connected hardware device does not support audio location streaming or does not support room layout capabilities.
11001 – Device not connected	No audio device is connected.
11006 – Room info unavailable	Room info unavailable.
11007 – Bar locations unavailable	Bar locations unavailable.

Sample requests

1. Valid request/response

Request

JSON

```
{
  "request": "v1/room/layout",
  "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
  "clientId": "test",
}
```

Response

JSON


```
{
    "request": "v1/room/layout",
    "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
    "clientId": "test",
    "body": {
        "roomDimensions": {
            "x": 16764,
            "y": 10668
        },
        "bars": [
            {
            "wall": "South",
            "location": {
                "x": 8382,
                "y": 0
            }
        },
        {
            "wall": "North",
            "location": {
                "x": 11382,
                "y": 10002
            }
        }
    ]
}
```

2. Unsupported device

This is a valid request but returns an error because the device doesn't support room layout capabilities.

Request

JSON

```
{
   "request": "v1/room/layout",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": "test"
}
```

Response

JSON

```
{
   "request": "v1/room/layout",
   "requestId": "3abe203s-42b7-4b0b-9awaf-5c381793a192",
   "clientId": " test",
    "errors": [
      {
            "statusCode": 11000,
            "message": "Unsupported device"
      }
   ]
}
```

v1/room/zones

Get room zones information

Only HDL410 systems support this feature.

Returns the zone information of the connected Nureva® audio conferencing system's room.

A zone is an area, up to the size of the whole room, that has a label and a type. A room is composed of 0 or more user defined zones. Zones of the same type may not overlap but zones of different types may overlap, and all zones must be entirely contained within a room.

When a sound location event occurs within a room, the sound can trigger the 1 or more zones that encompass the sound location. It is possible for a single sound location event to trigger 0 or more zones, but as stated above, if multiple zones are triggered by the same sound location event, they would need to be zones of different types.

Zones are configurable by end users through Nureva Console software.

Learn more about zones

Request

JSON

```
{
  "request": "v1/room/zones",
  "requestId": string,
  "clientId": string
}
```

Response

JSON

```
{
    "request": "v1/room/zones",
    "requestId": string,
    "clientId": string,
    "body": {
        "zones": {
            "types": ["Switching"]
            "geometry": {
                "point1": {
                    "x": number,
                    "y": number,
                },
                "point2": {
                    "x": number,
                    "y": number,
                },
            },
            "label": string,
            "id": string,
        } [ ]
    }
}
```

types	An array of 1 or more strings defining the purpose of this zone. Presently, the only possible type of zone is "Switching" indicating that this zone is intended to define support for camera switching within this room. It is not possible for the same type of zone to appear more than once within this array.
geometry	Position of the zone in room coordinates, as defined in Get room layout information, where the southwest corner of the room is taken as the origin (0,0) of the coordinate system. All zones are rectangular or square with their bounding box defined by the point1 and point2 properties on opposite corners. • **point1**: The **x** and **y** location, in room coordinates, of one corner of the zones' bounding box in millimeters. **point1** is the opposite corner to **point2**. • **point2**: The **x** and **y** location, in room coordinates, of one corner of the zones' bounding box in millimeters. **point2** is the opposite corner to **point1**.
label	The meaningful name given to the zone when defined through Nureva Console. This identifier should be human readable.
id	A unique identifier for the zone that is not intended to be human readable.

Errors

Status code	Description
10000 – Bad request	Request does not conform to the specification. The message field will include detailed information about which part of the request is incorrect.
11000 – Unsupported device	Connected hardware device does not support audio location streaming.
11001 – Device not connected	No audio device is connected.
11006 – Room info unavailable	Room information unavailable.
11008 – Zone information unavailable	Zone information unavailable.

Sample requests

1. Valid request/response

Request

JSON

```
{
  "request": "v1/room/zones",
  "requestId": "1234",
  "clientId": "1111",
}
```

Response

JSON


```
{
    "request": "v1/room/zones",
    "requestId": "1234",
    "clientId": "1111",
    "body": {
        "zones": [
            {
                "types": ["Switching"],
                "geometry": {
                    "point1": {
                        "x": 0,
                        "y": 0
                    },
                    "point2": {
                        "x": 4567,
                        "y": 7294
                    },
                },
                "label": "Front of room",
                "id": "1b9d6bcd-bbfd-4b2d-9b5d-ab8dfbbd4bed"
            }
        ]
    }
}
```

2. Unsupported device

This is a valid request. However, in this fictitious example, the device is an HDL300 which doesn't support zones capabilities so it returns an error.

Request

JSON


```
{
   "request": "v1/room/zones",
   "requestId": "1234",
   "clientId": "1111",
}
```


Response

JSON


```
{
   "request": "v1/room/zones",
   "requestId": "1234",
   "clientId": "1111",
    "errors": [
      {
            "statusCode": 11000,
            "message": "Unsupported device"
      }
   ]
}
```