OPEN API – TCP Protocol

Rev 6. 05-19-2022

Introduction

The following guide outlines the TCP Protocol including commands that are available for all Dante Connect Series and Network Connect Series models. There are more advanced commands available which are not listed. Please email techsupport@leaprofessional.com for assistance with advanced API integration.

Nomenclature

element - an item in the amplifier than can be controlled and/or monitored
control - term applied to an element that can be controlled and monitored, i.e. read/write
sensor - term applied to an element that can be monitored only (i.e. read-only)
object - a logical grouping of elements (see websocket API doc)

Supported commands (case-insensitive)

GET <object/element> - get an element's value (control or sensor); normal response = element's value
SET <object/element> <value> - set an element's value (controls only); normal response = OK
SUBSCRIBE <object/element>- sign up for notifications of changes to an element's value (control or sensor); normal response = element's value
UNSUBSCRIBE <object/element> - cancel a subscription; normal response = OK

Details

a) uses TCP port 4321

b) Any LEA Connect Firmware version 1.1.0.X and higher supports TCP Communication. Firmware versions prior to this only support websocket protocol.

b) commands and responses are newline (i.e. '\n', 0x0a) delimited <-- THIS IS IMPORTANT! All messages end with '\n'

c) all commands generate response messages

d) subscriptions generate additional, asynchronous messages (of course, subscriptions require a persistent connection)

e) illegal commands generate error messages (i.e. response beginning with 'error')

f) out-of-range numeric values will be clamped (i.e. no error response; this is the same behavior as websocket API)

g) string/enum values containing spaces must be enclosed in double-quotes

Example commands with responses

a) To get Dante On Ramp Channel 1 setting:
Command: get /amp/channels/1/inputSelector/danteOnRamp\n
Amp Response: /amp/channels/1/inputSelector/danteOnRamp "Post Crossover"\n

b) Set channel 1 primary input fader level to -3.0dB:
Command: set /amp/channels/1/inputSelector/primaryFader -3.0\n
Amp Response: OK\n

c) Subscribe to channel 1 input level:
Command: subscribe /amp/channels/1/levels/level_db\n
Amp Response: /amp/channels/1/levels/level_db -52.78131103515625\n

d) Unsubscribe to channel 1 input level:
Command: unsubscribe /amp/channels/1/levels/level_db\n
Amp Response: OK\n

e) If a command is sent that the amplifier does not recognize:
Command Sent: set /amp/channels/1/inputSelector/danteOnRamp "Post Toasties"\n
Amp Response: error: unsupported value: Post Toasties (/amp/channels/1/inputSelector/danteOnRamp)\n

f) If a command is sent that is not in the proper syntax/format:
Command: get something-that-doesn't-exist\n
Amp Response: error: cannot parse - 'get something-that-doesn't-exist'\n

g) If you forget to put \n at the end of your command:
Command: message that's missing delimiter
Amp Response: (no response)

Security

All TCP communication to and from the LEA Connect Series Amplifier is on port 4321. The amplifier will only send and receive API commands for the amplifier's control and monitoring.

Example Sessions

Subscribe /amp/channels/1/levels/level_db
/amp/channels/1/levels/level_db -52.78131103515625
/amp/channels/1/levels/level_db -53.08610916137695
/amp/channels/1/levels/level_db -52.2020378112793
/amp/channels/1/levels/level_db -52.78131103515625
/amp/channels/1/levels/level_db -51.658992767333984
/amp/channels/1/levels/level_db -52.2020378112793

/amp/channels/1/levels/level_db -51.92626953125
/amp/channels/1/levels/level_db -51.658992767333984
/amp/channels/1/levels/level_db -51.92626953125
/amp/channels/1/levels/level_db -53.40199279785156
/amp/channels/1/levels/level_db -52.2020378112793
/amp/channels/1/levels/level_db -52.486846923828125
/amp/channels/1/levels/level_db -51.92626953125
/amp/channels/1/levels/level_db -52.486846923828125
/amp/channels/1/levels/level_db -53.08610916137695
/amp/channels/1/levels/level_db -52.2020378112793
/amp/channels/1/levels/level_db -53.08610916137695
/amp/channels/1/levels/level_db -51.658992767333984
/amp/channels/1/levels/level_db -52.78131103515625
/amp/channels/1/levels/level_db -53.08610916137695

Subscribe /amp/channels/1/levels/level_db
/amp/channels/2/levels/level_db -53.40199279785156
/amp/channels/1/levels/level_db -52.78131103515625
/amp/channels/2/levels/level_db -54.794715881347656
/amp/channels/1/levels/level_db -51.92626953125
/amp/channels/2/levels/level_db -53.72980499267578
/amp/channels/1/levels/level_db -52.78131103515625
/amp/channels/2/levels/level_db -54.425048828125
/amp/channels/2/levels/level_db -54.794715881347656
/amp/channels/1/levels/level_db -51.39969253540039
/amp/channels/2/levels/level_db -53.72980499267578
/amp/channels/1/levels/level_db -51.658992767333984
/amp/channels/1/levels/level_db -52.2020378112793
/amp/channels/2/levels/level_db -54.070472717285156
/amp/channels/1/levels/level_db -53.08610916137695
/amp/channels/2/levels/level_db -55.180816650390625
/amp/channels/1/levels/level_db -52.2020378112793
/amp/channels/2/levels/level_db -54.425048828125
/amp/channels/1/levels/level_db -52.78131103515625
/amp/channels/2/levels/level_db -54.794715881347656
/amp/channels/1/levels/level_db -52.486846923828125
/amp/channels/2/levels/level_db -54.425048828125
/amp/channels/1/levels/level_db -52.78131103515625
/amp/channels/2/levels/level_db -54.794715881347656
/amp/channels/1/levels/level_db -52.486846923828125
/amp/channels/2/levels/level_db -54.425048828125
/amp/channels/1/levels/level_db -51.92626953125
/amp/channels/2/levels/level_db -54.070472717285156

/amp/channels/1/levels/level_db -52.2020378112793
/amp/channels/2/levels/level_db -54.425048828125
/amp/channels/1/levels/level_db -52.486846923828125
/amp/channels/1/levels/level_db -53.08610916137695
/amp/channels/2/levels/level_db -55.180816650390625
/amp/channels/1/levels/level_db -51.92626953125

unsubscribe /amp/channels/1/levels/level_db
/amp/channels/1/levels/level_db -52.78131103515625
/amp/channels/1/levels/level_db -52.2020378112793
/amp/channels/2/levels/level_db -54.070472717285156
OK
/amp/channels/2/levels/level_db -54.794715881347656
/amp/channels/2/levels/level_db -54.070472717285156
/amp/channels/2/levels/level_db -54.794715881347656
/amp/channels/2/levels/level_db -53.72980499267578
/amp/channels/2/levels/level_db -55.180816650390625
/amp/channels/2/levels/level_db -54.070472717285156

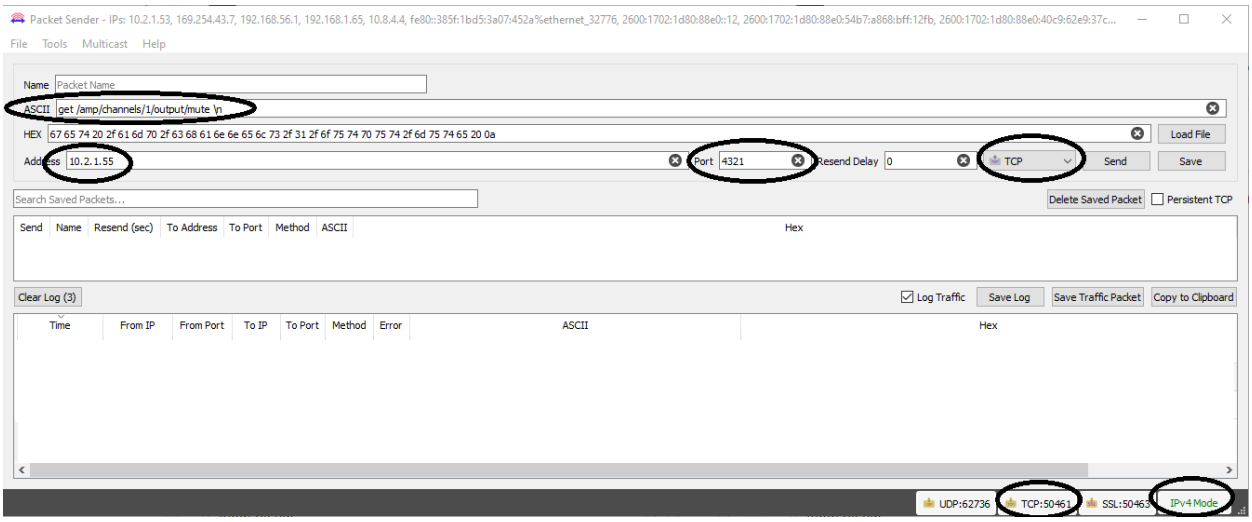
unsubscribe /amp/channels/2/levels/level_db
/amp/channels/2/levels/level_db -55.58488464355469
OK
subscribe /amp/channels/1/inputSelector/danteOnRamp
/amp/channels/1/inputSelector/danteOnRamp "Analog Input"
set /amp/channels/1/inputSelector/danteOnRamp "Post Crossover"
OK
/amp/channels/1/inputSelector/danteOnRamp "Post Crossover"
get /amp/channels/1/inputSelector/danteOnRamp
/amp/channels/1/inputSelector/danteOnRamp "Post Crossover"
get /amp/channels/2/inputSelector/danteOnRamp
/amp/channels/2/inputSelector/danteOnRamp "Analog Input"
set /amp/channels/1/inputSelector/danteOnRamp "Post Toasties"
error: unsupported value: Post Toasties (/amp/channels/1/inputSelector/danteOnRamp)
set /amp/channels/1/inputSelector/primaryFader -80.0
OK
/amp/channels/1/inputSelector/primaryFader -80.0

Examples with Packet Sender

Packet Sender is a free open source cross-platform application that is widely used to send and receive TCP messages on a specified port. A free download is available at http://www.packetsender.com.
Below are some screen shots to help you use packet sender successfully and several examples.

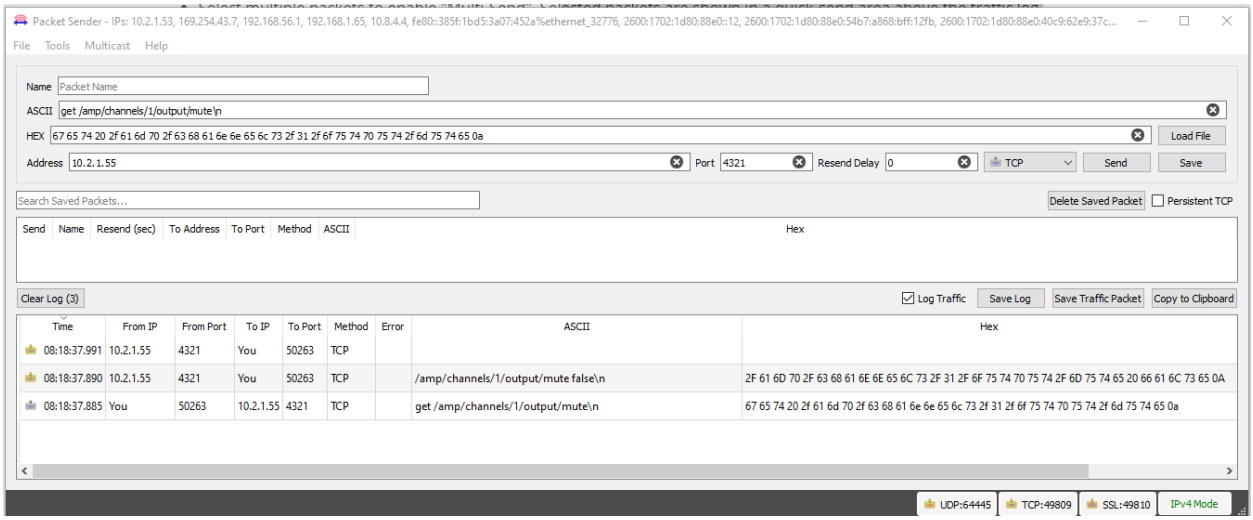
Packet Sender Setup

- In the bottom right corner, make sure it is in IPv4 Mode (not IPv6 Mode)
- In the bottom right corner, make sure TCP is enabled
- Enter the Port number "4321"
- On the same line select TCP from the dropdown list, should be TCP by default
- Enter the IP address of the amplifier to receive the message
- Enter the ASCII message
    - get/set/subscribe/unsubscribe
    - url of the element
    - value if method is set
    - All messages must be terminated with a new line "\n"
- Press "Send" button when finished



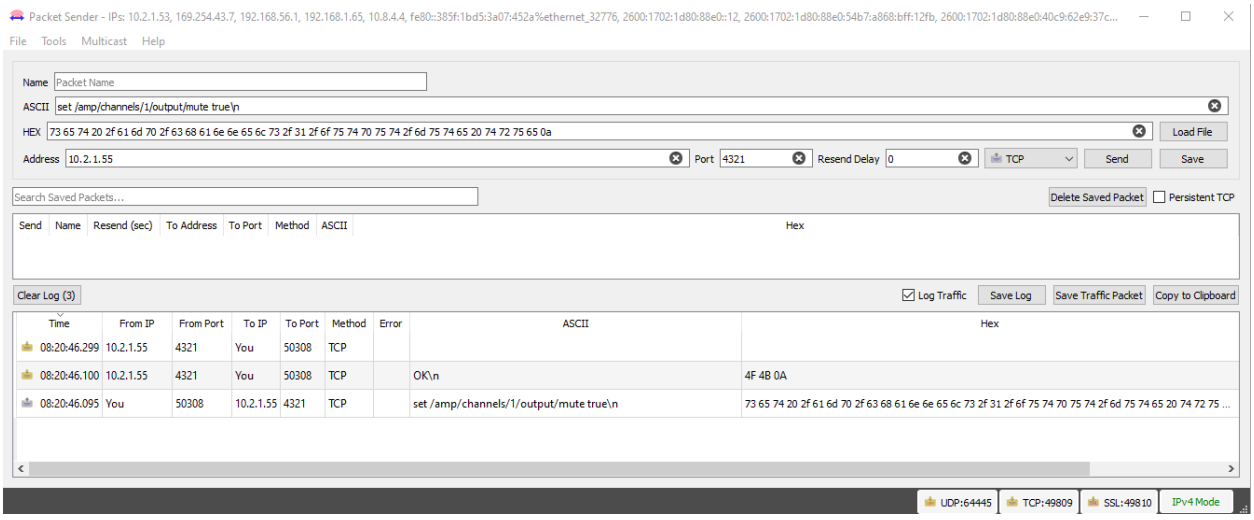
Example: Get Ch1 Mute

Command: get /amp/channels/1/output/mute\n



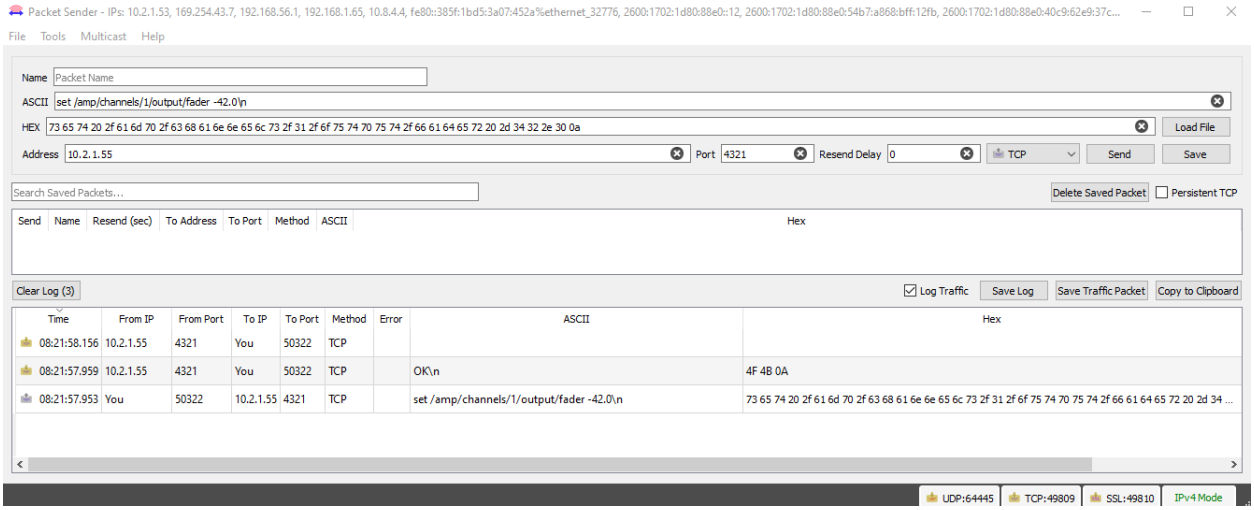
Example: Set Ch1 Mute

Command: set /amp/channels/1/output/mute true\n



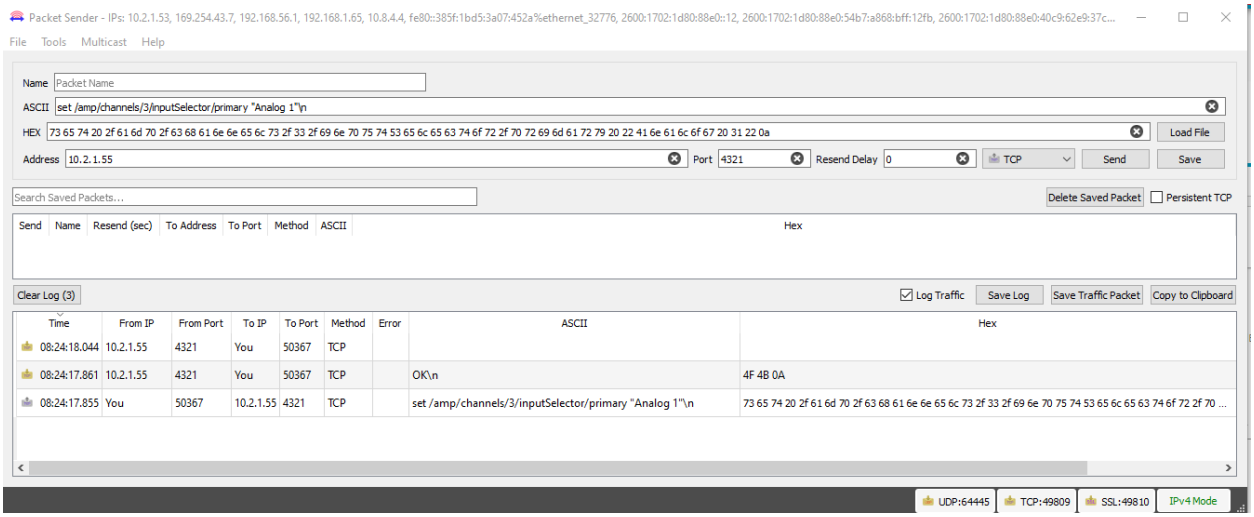
Example: Set Ch2 output Fader to -42.0dB

Command: set /amp/channels/1/output/fader -42.0\n



Example: Set Ch3 Input Selector to Analog 1

Command: set /amp/channels/3/inputSelector/primary "Analog 1"\n



Supported objects/elements (case-sensitive)

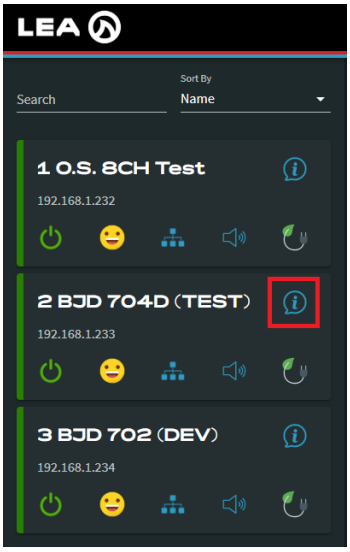
Object URL Scheme

Below is a list of available base object URL destinations
"x" is the desired channel number
"#" is the desired input number
"*" is the desired filter number

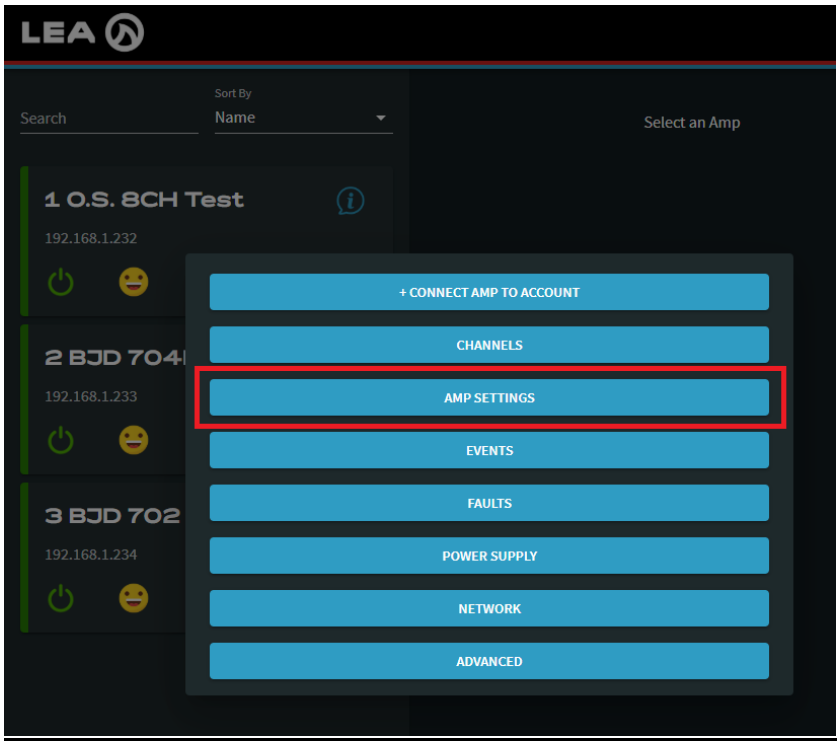
/amp/deviceInfo/
/amp/powerSupply/
/amp/autoStandby/
/amp/signalGenerator/
/amp/inputs/analog/#/
/amp/channels/x/inputSelector/
/amp/channels/x/crossover/
/amp/channels/x/outputEqFilters/*/
/amp/channels/x/rmsLimiter/
/amp/channels/x/peakLimiter/
/amp/channels/x/output/
/amp/channels/x/levels/
/amp/channels/x/loadMonitor/

Amplifier Device Info

Click on the amplifier info button



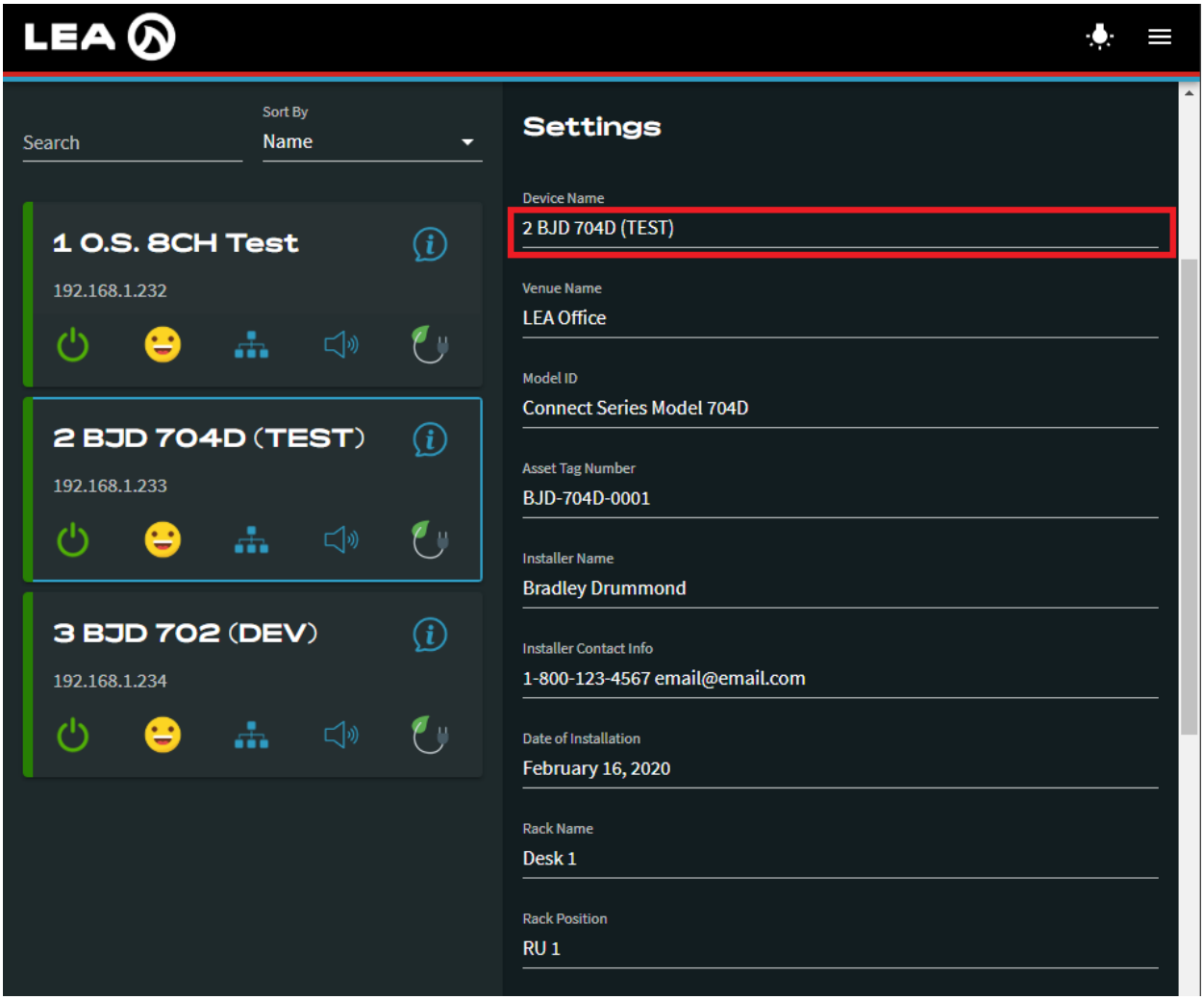
Click on Amp Settings to view the relevant page for this API section



Device Name

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/deviceInfo/deviceName
Values: any text up to 64 characters
Example: get /amp/deviceInfo/deviceName\n

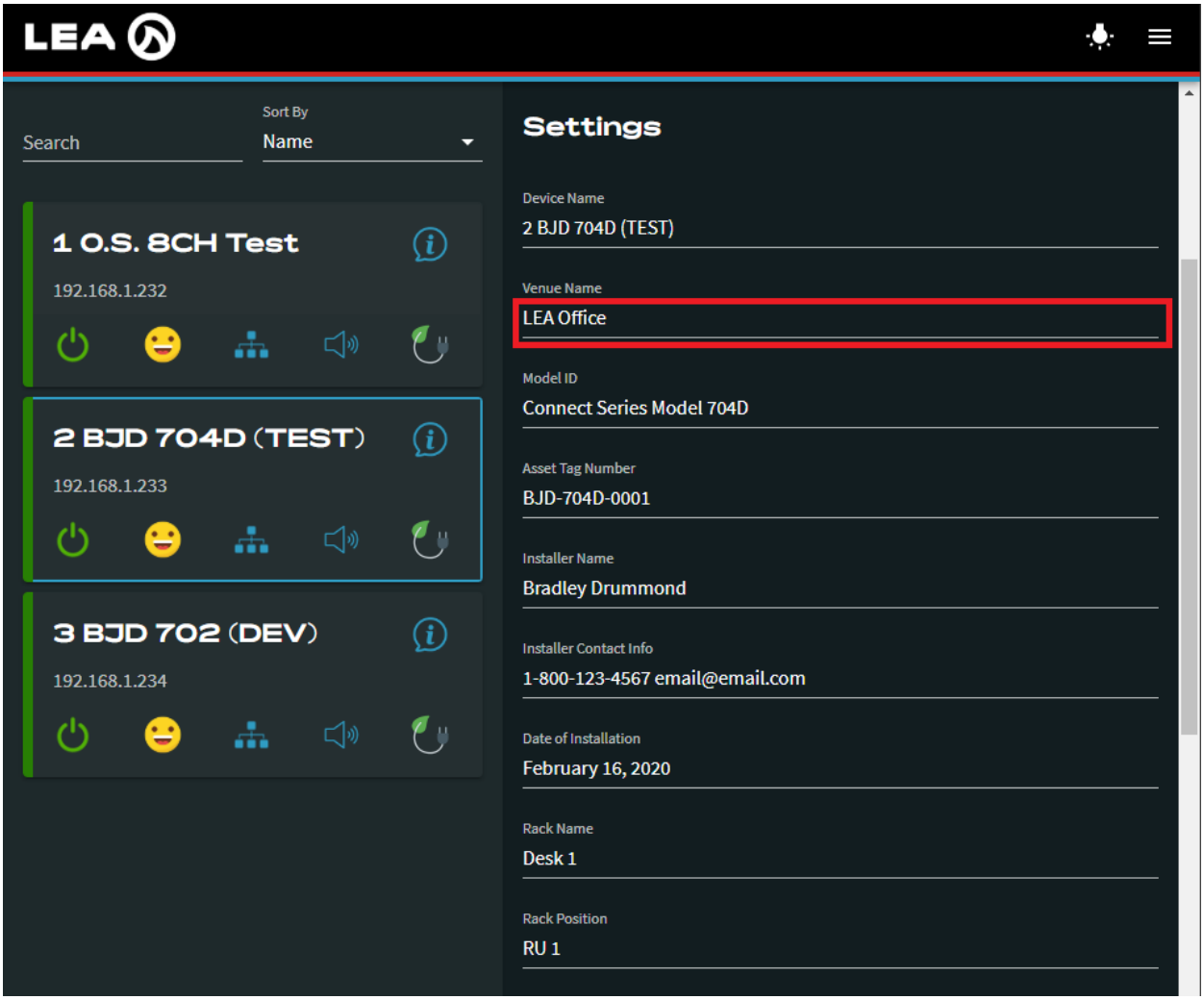
- Response: /amp/deviceInfo/deviceName "2 BJD 704D (TEST)"\n
- The command asked for the amplifiers device name and got the response 2 BJD 704D (TEST)



Venue Name

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/deviceInfo/venueName
Values: any text up to 64 characters
Example: get /amp/deviceInfo/venueName\n

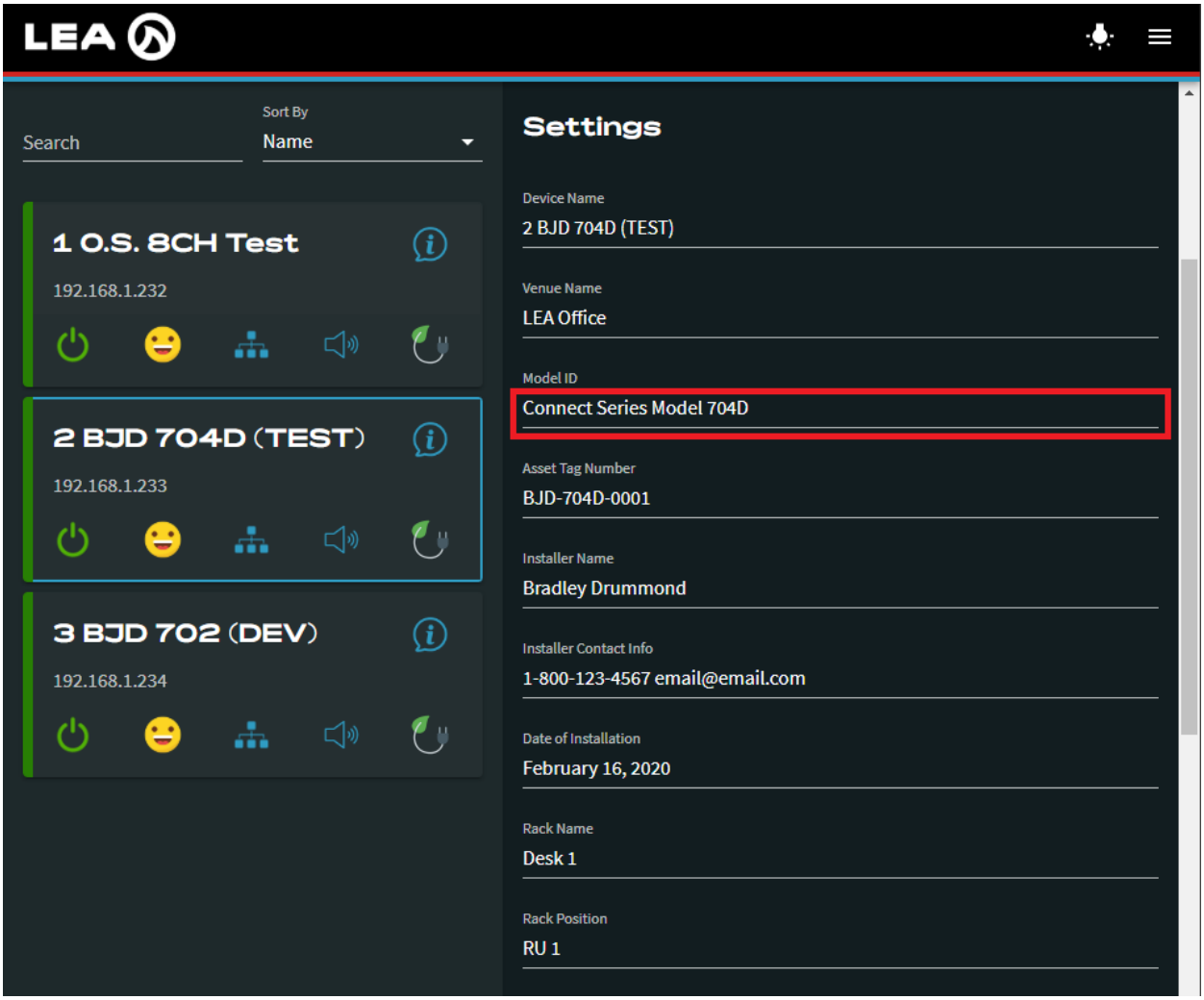
- Response: /amp/deviceInfo/venueName "LEA Office"\n
- The command asked for the amplifiers device name and got the response LEA Office



Model ID

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/modelID
Values: Amplifier Model
Example: get /amp/deviceInfo/modelID\n

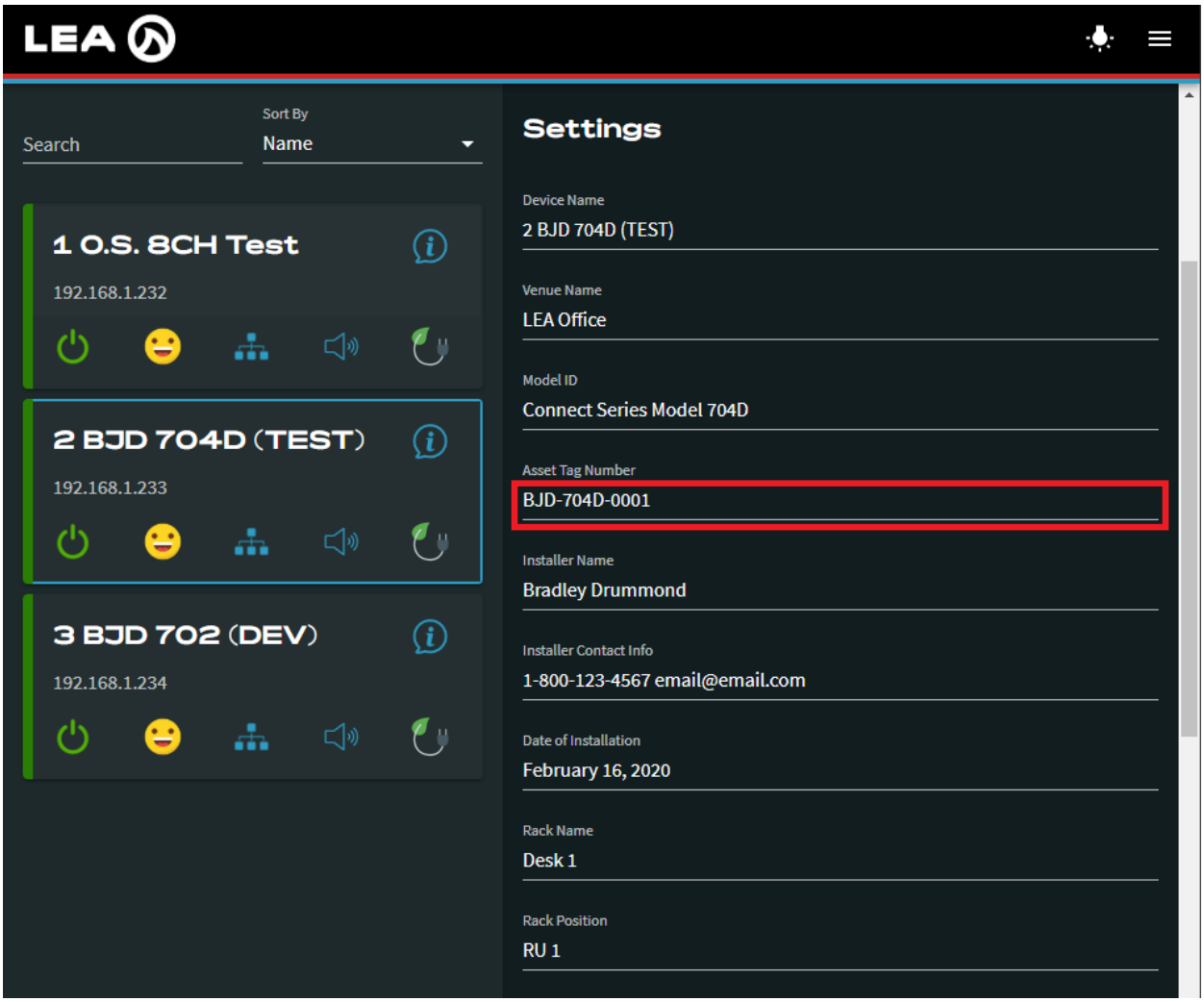
- Response: /amp/deviceInfo/modelID "Connect Series Model 704D"\n
- The command asked for the amplifiers model ID and got the response Connect Series Model 704D



Asset Tag Number

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/deviceInfo/assetTagNumber
Values: any text up to 64 characters
Example: get /amp/deviceInfo/assetTagNumber\n

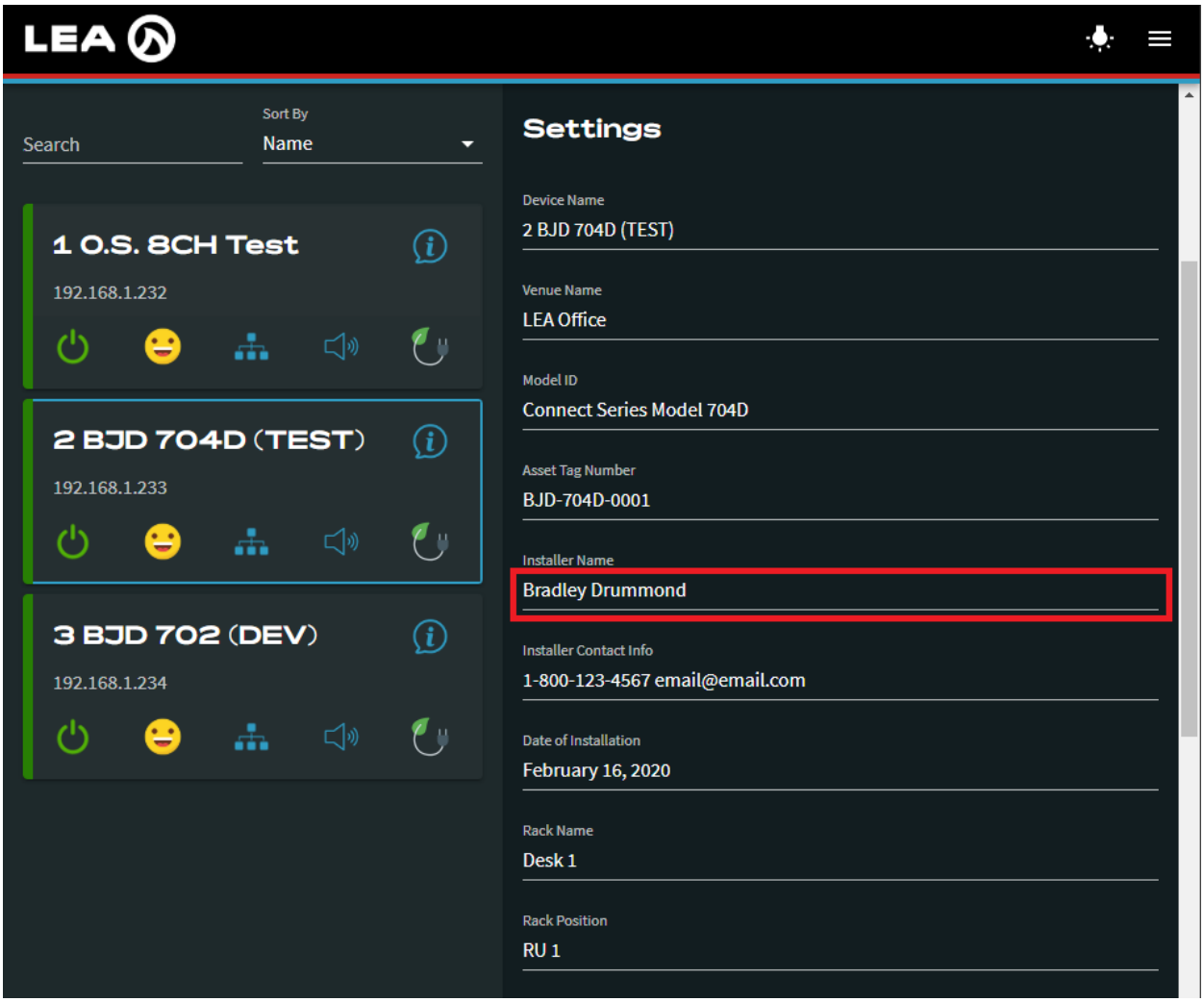
- Response: /amp/deviceInfo/assetTagNumber "BJD-704D-0001"\n
- The command asked for the amplifiers asset tag number and got the response BJD-704D-0001



Installer Name

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/deviceInfo/installerName
Values: any text up to 64 characters
Example: get /amp/deviceInfo/installerName\n

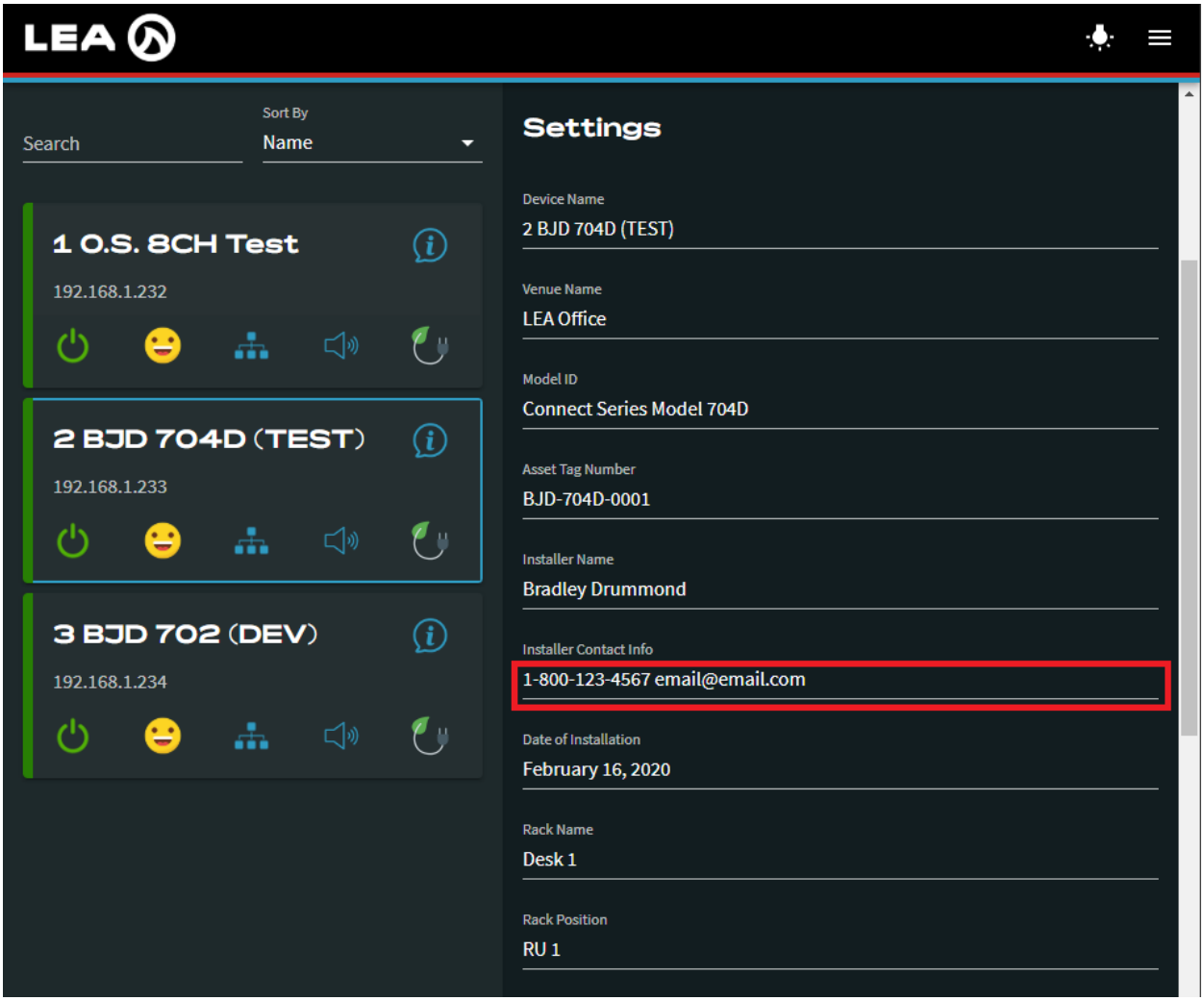
- Response: /amp/deviceInfo/installerName "Bradley Drummond"\n
- The command asked for the amplifiers installer name and got the response Bradley Drummond



Installer Contact Info

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/deviceInfo/installerContactInfo
Values: any text up to 64 characters
Example: get /amp/deviceInfo/installerContactInfo\n

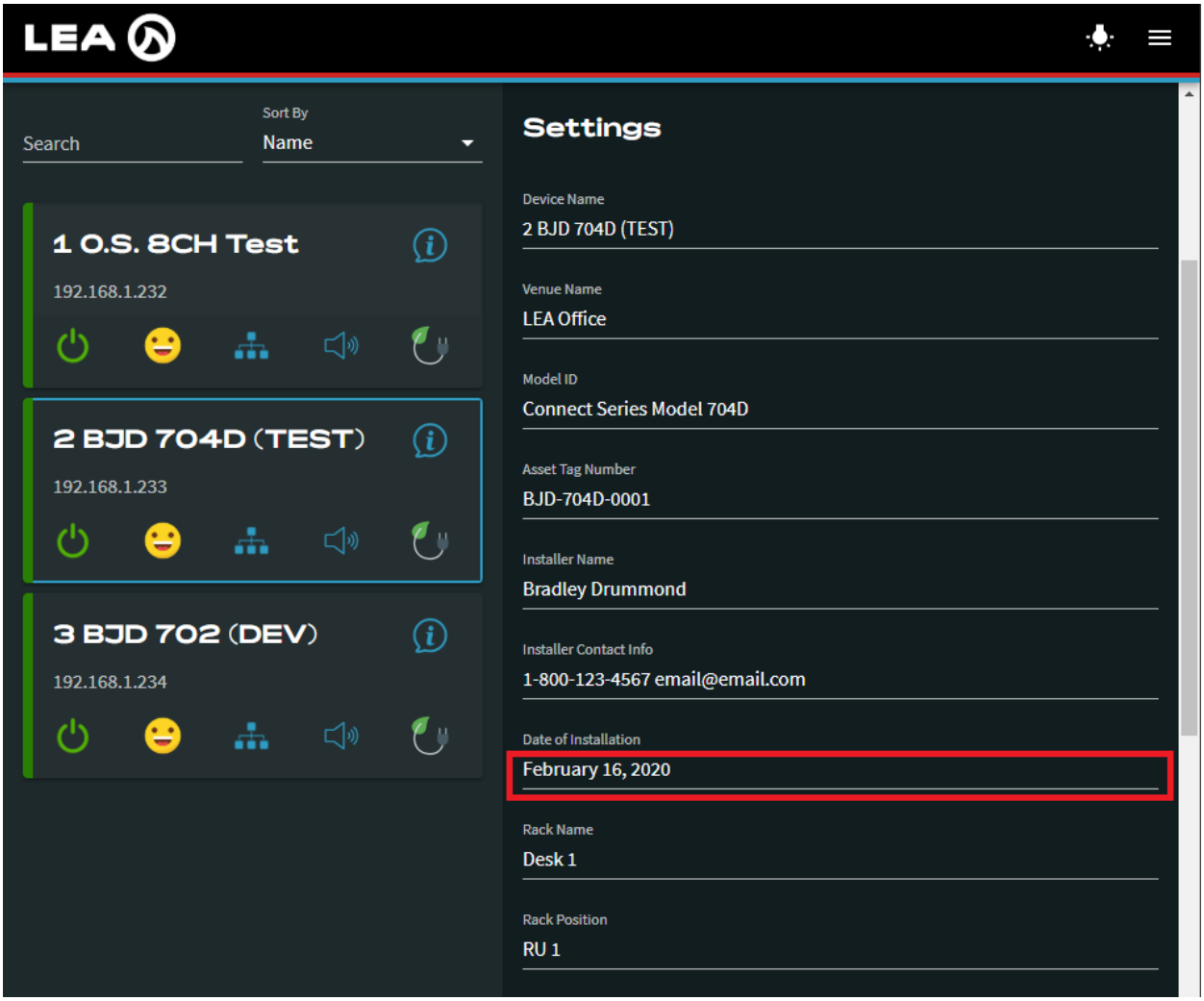
- Response: /amp/deviceInfo/installerContactInfo "1-800-123-4567 email@email.com"\n
- The command asked for the amplifier's installer contact info and got the response 1-800-123-4567 email@email.com



Date of Installation

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/deviceInfo/dateOfInstallation
Values: any text up to 64 characters
Example: get /amp/deviceInfo/dateOfInstallation\n

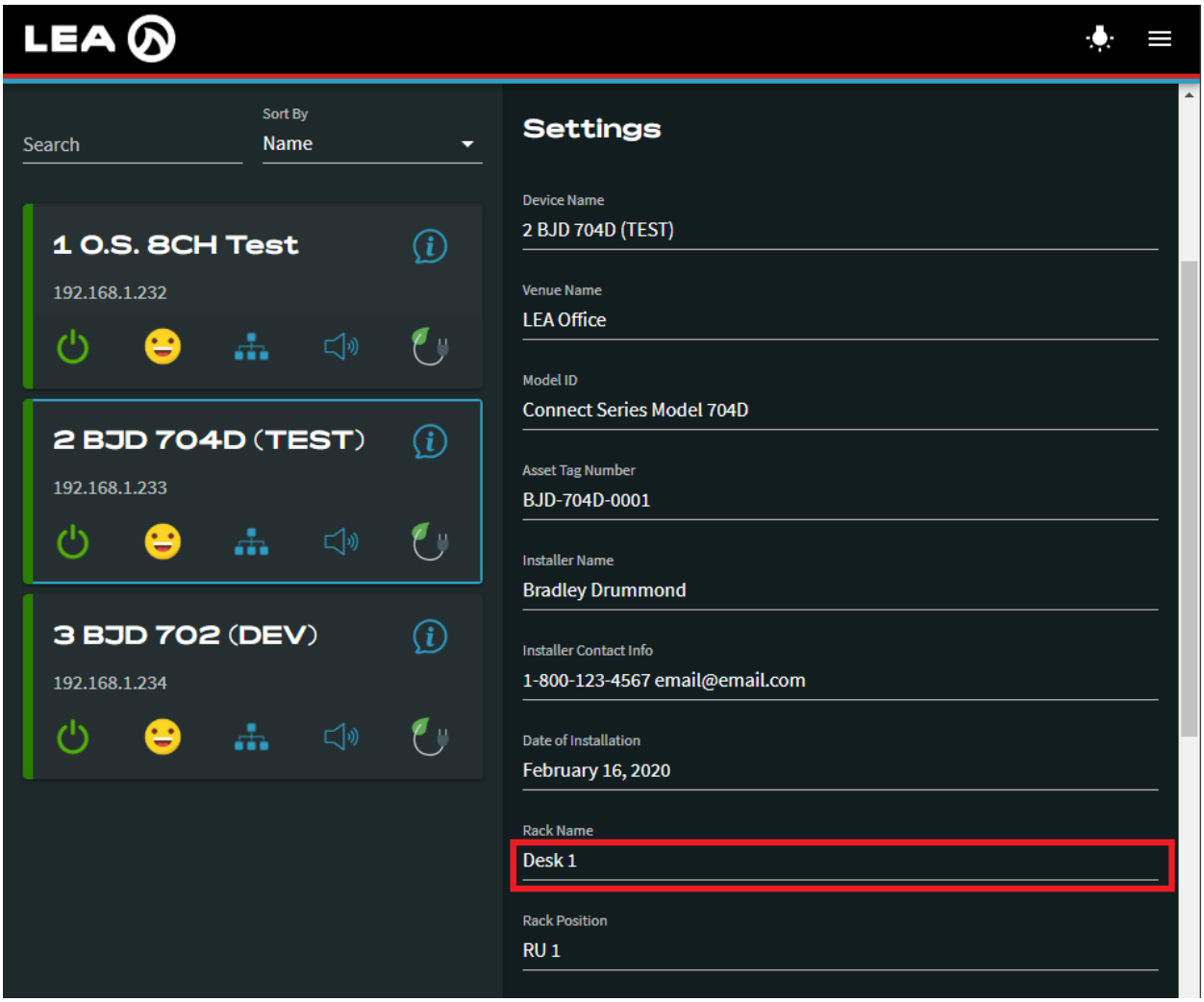
- Response: /amp/deviceInfo/dateOfInstallation "2020-02-16T20:47:00.000Z"\n
- The command asked for the amplifiers asset tag number and got the response February 16, 2020 time 20:47:00.000



Rack Name

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/deviceInfo/rackName
Values: any text up to 64 characters
Example: set /amp/deviceInfo/rackName "Rack 1"\n

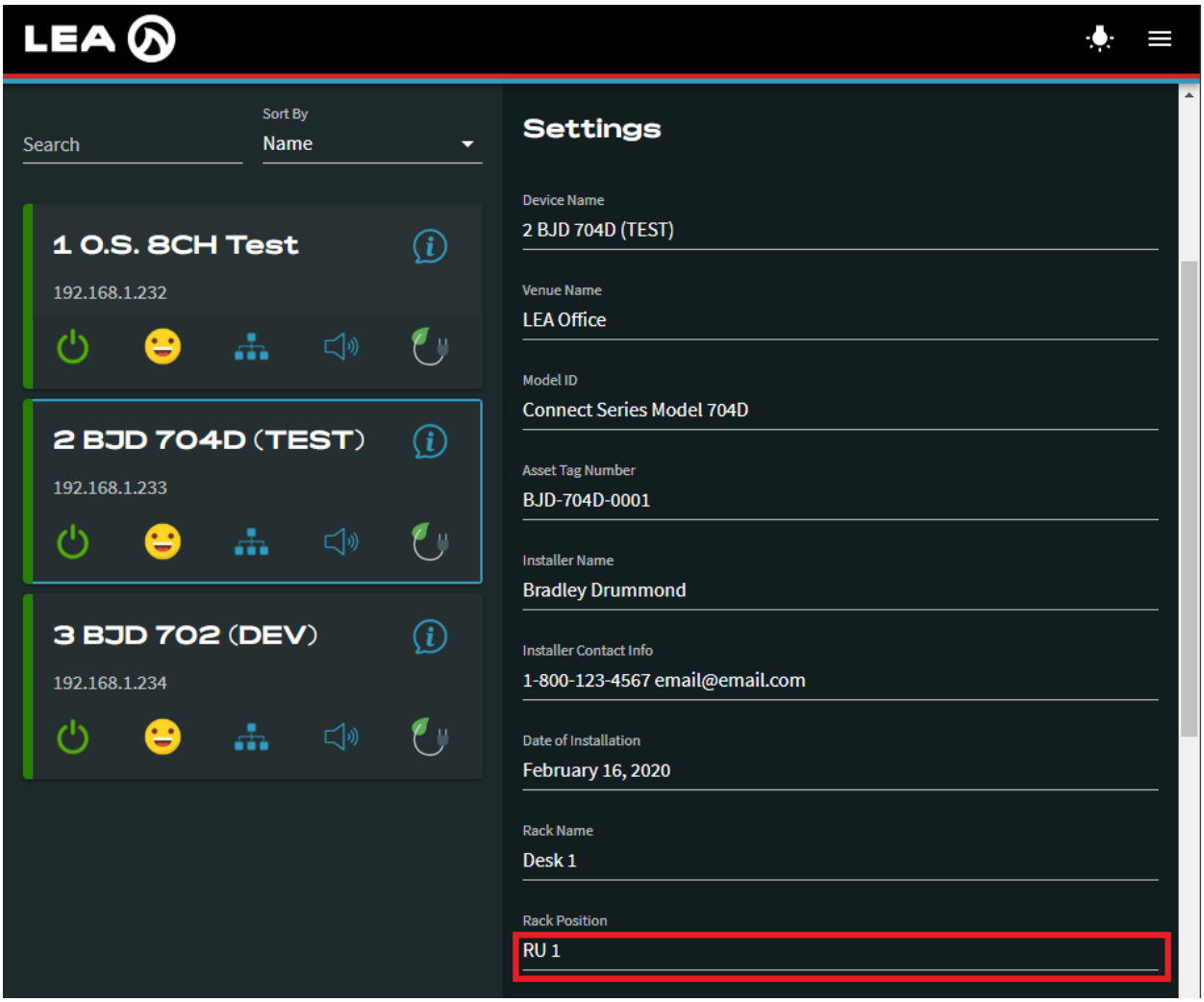
- Response: OK \n
- The command successfully set the amplifiers rack name to Rack 1



Rack Position

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/deviceInfo/rackPosition
Values: any text up to 64 characters
Example: get /amp/deviceInfo/rackPosition\n

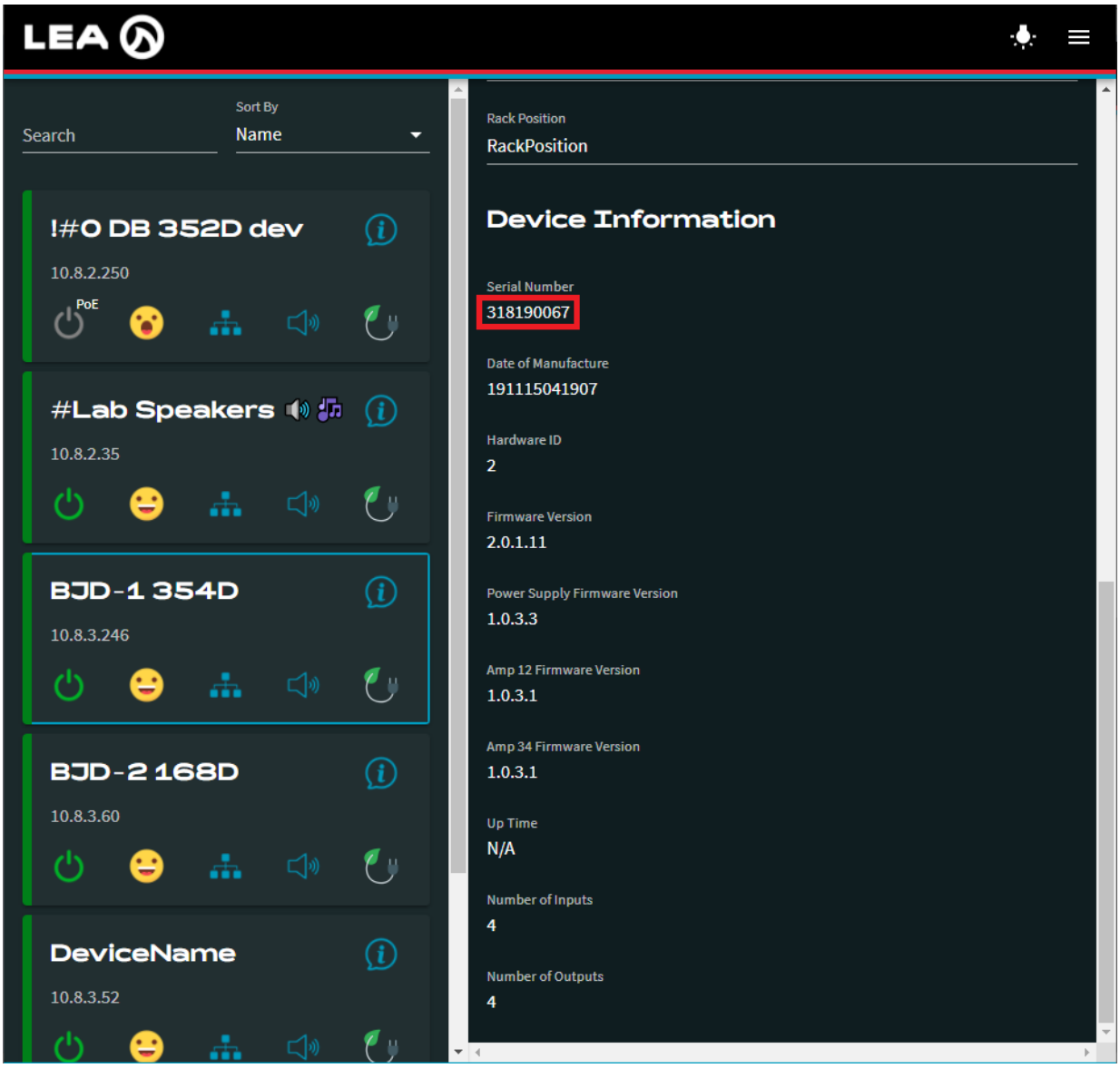
- Response: /amp/deviceInfo/rackPosition "RU 1"\n
- The command asked for the amplifiers rack position and got the response RU 1



Serial Number

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/serialNumber
Values: any text up to 64 characters
Example: get /amp/deviceInfo/serialNumber\n

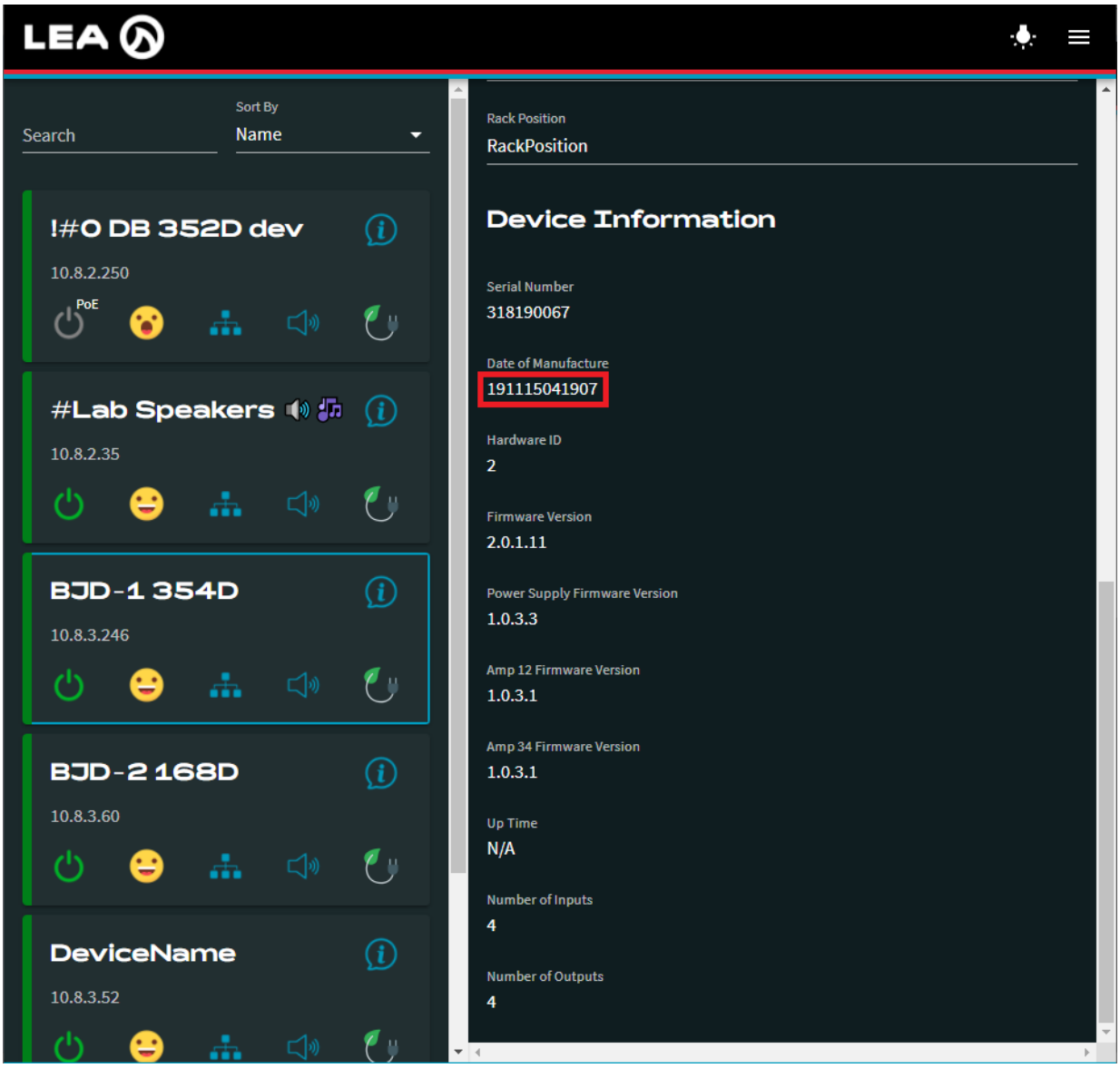
- Response: /amp/deviceInfo/serialNumber "31890067"\n
- The command asked for the amplifiers serial number and got the response 31890067



Date of Manufacture

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/dateOfMfg
Values: any text up to 64 characters
Example: get /amp/deviceInfo/ dateOfMfg \n

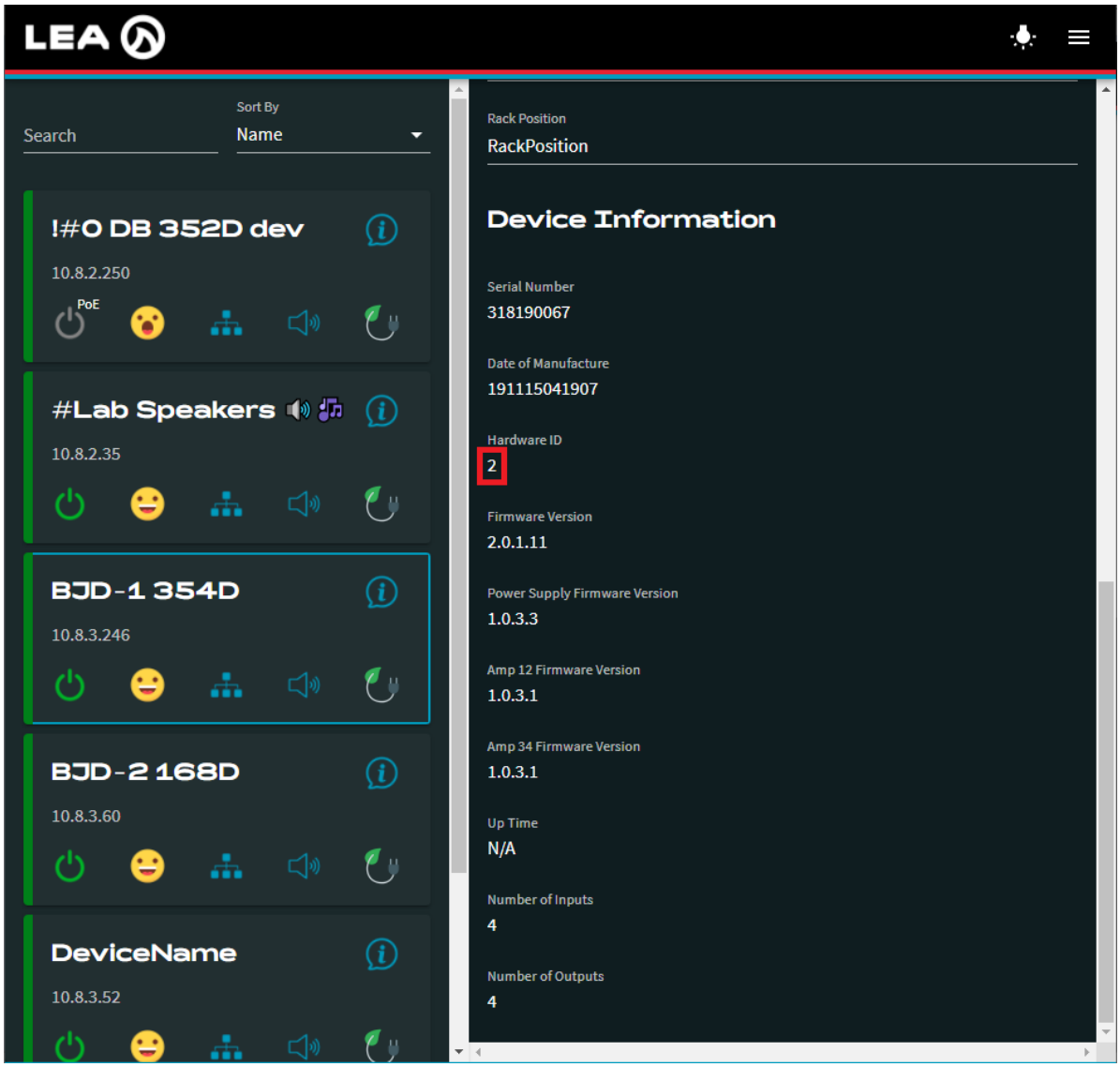
- Response: /amp/deviceInfo/ dateOfMfg "191115041907"\n
- The command asked for the amplifiers date of manufacture and got the response 191115041907



Hardware ID

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/hardwareID
Values: any text up to 64 characters
Example: get /amp/deviceInfo/hardwareID\n

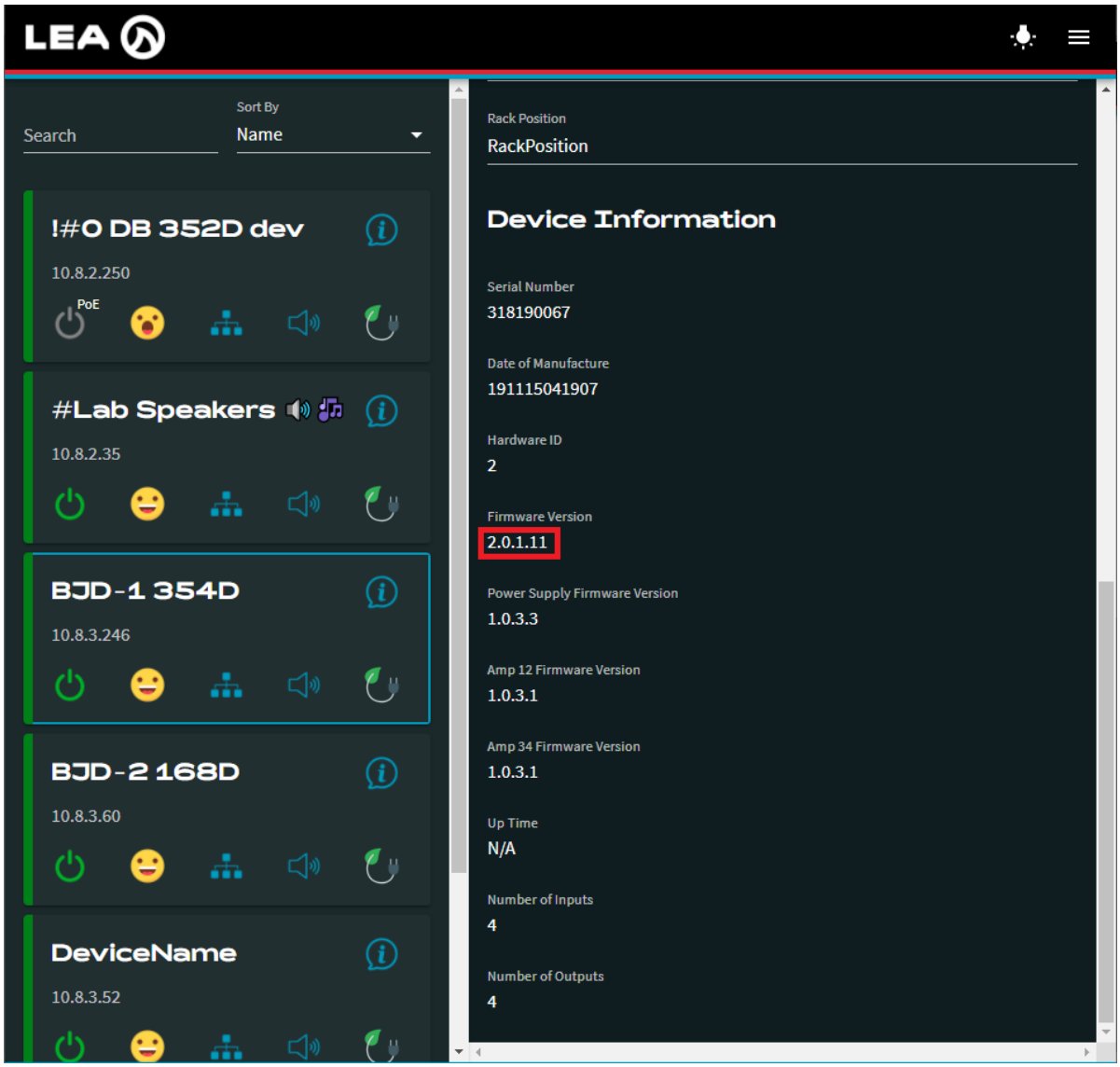
- Response: /amp/deviceInfo/hardwareID "2"\n
- The command asked for the amplifiers hardware ID and got the response 2



Firmware Version

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/firmwareVersion
Values: any text up to 64 characters
Example: get /amp/deviceInfo/firmwareVersion\n

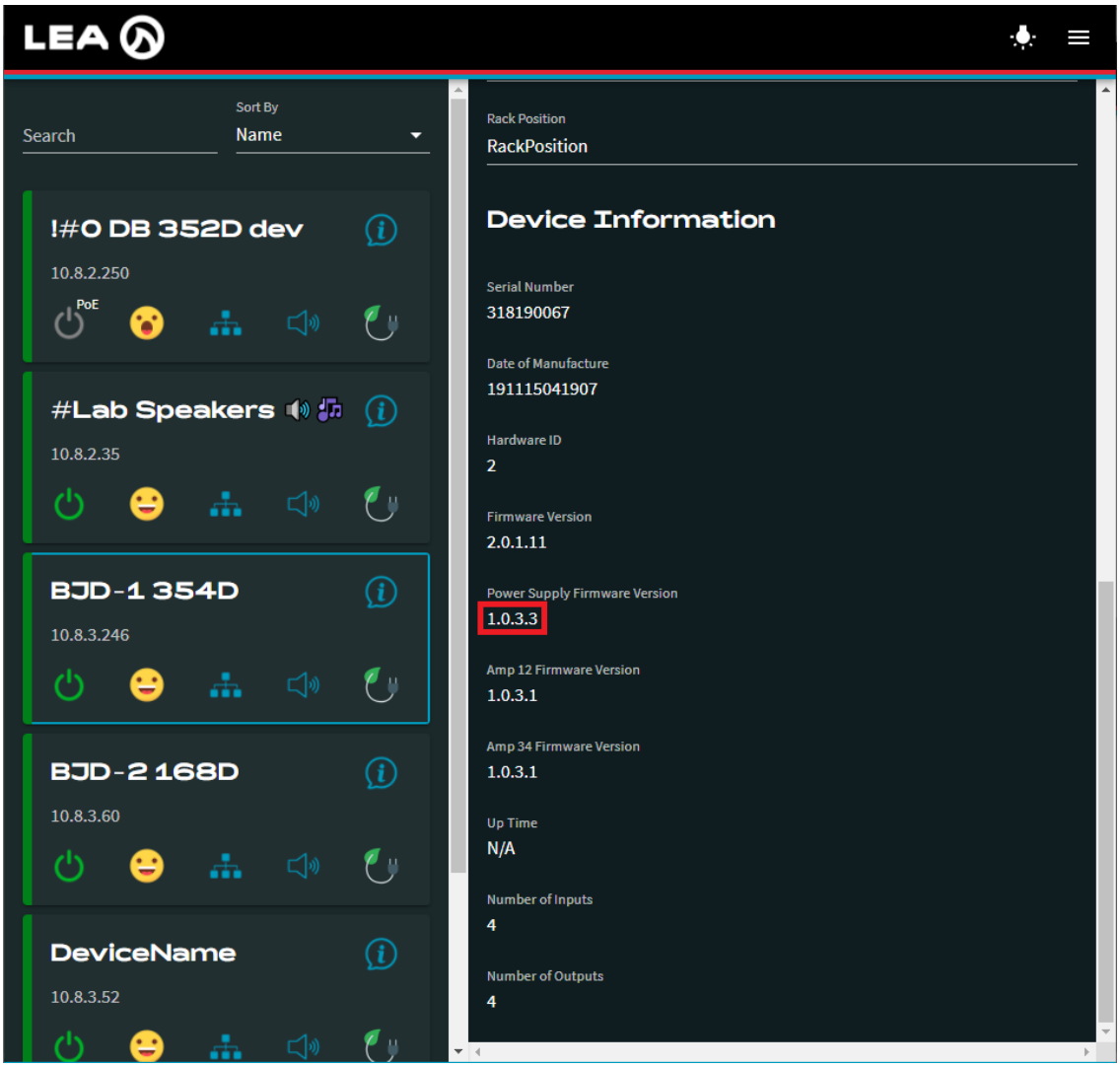
- Response: /amp/deviceInfo/ firmwareVersion "2.0.1.11"\n
- The command asked for the amplifiers firmware version and got the response 2.0.1.11



Power Supply Firmware Version

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/powerSupplyfirmwareVersion
Values: any text up to 64 characters
Example: get /amp/deviceInfo/powerSupplyfirmwareVersion\n

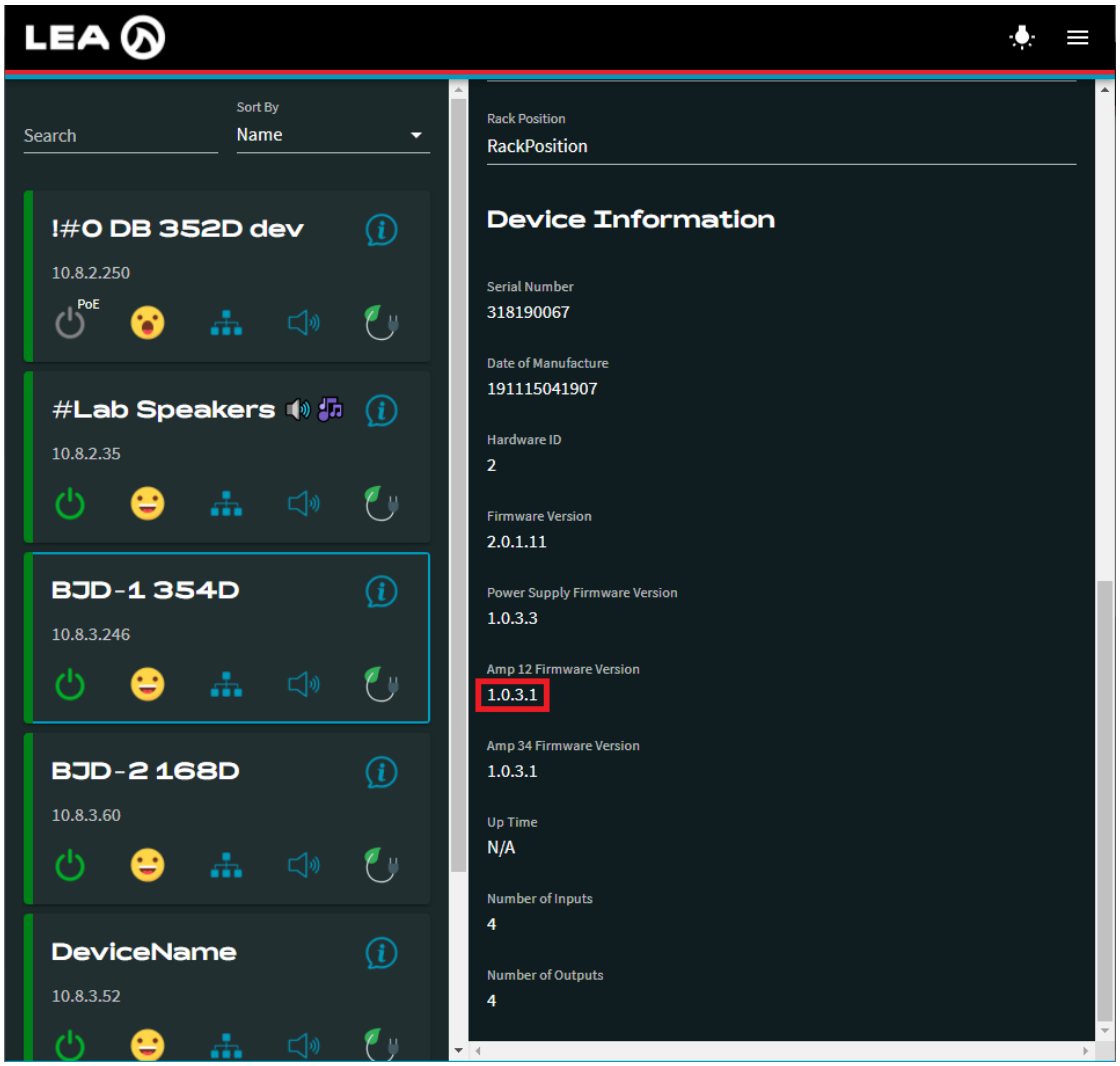
- Response: /amp/deviceInfo/powerSupplyfirmwareVersion "1.0.3.3"\n
- The command asked for the amplifiers power supply firmware version and got the response 1.0.3.3



Amp 12 Firmware Version

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/amp12firmwareVersion
Values: any text up to 64 characters
Example: get /amp/deviceInfo/amp12firmwareVersion\n

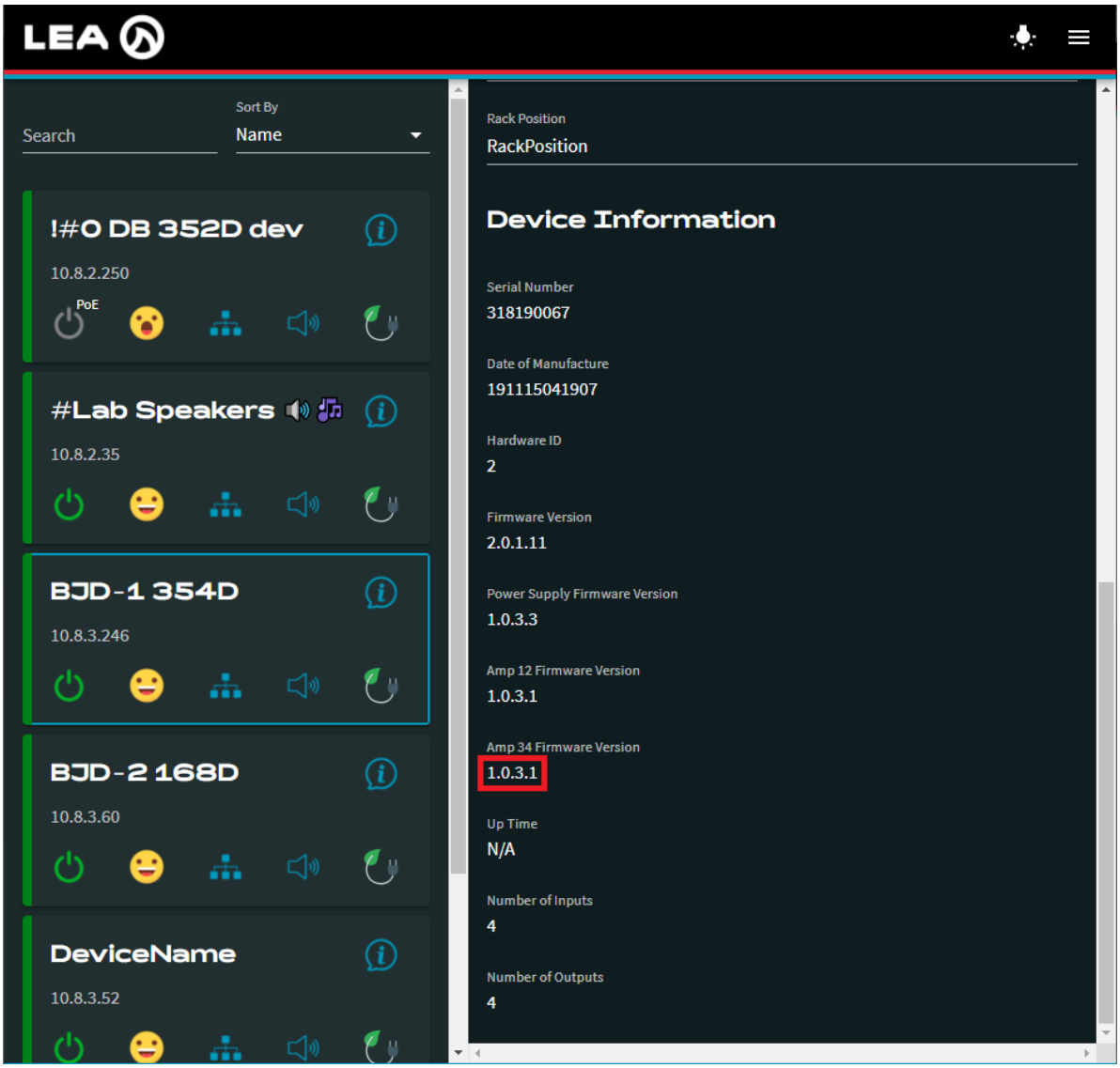
- Response: /amp/deviceInfo/amp12firmwareVersion "1.0.3.1"\n
- The command asked for the amplifiers amp 12 firmware version and got the response 1.0.3.1



Amp 34 Firmware Version

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/amp12firmwareVersion
Values: any text up to 64 characters
Example: get /amp/deviceInfo/amp34firmwareVersion\n

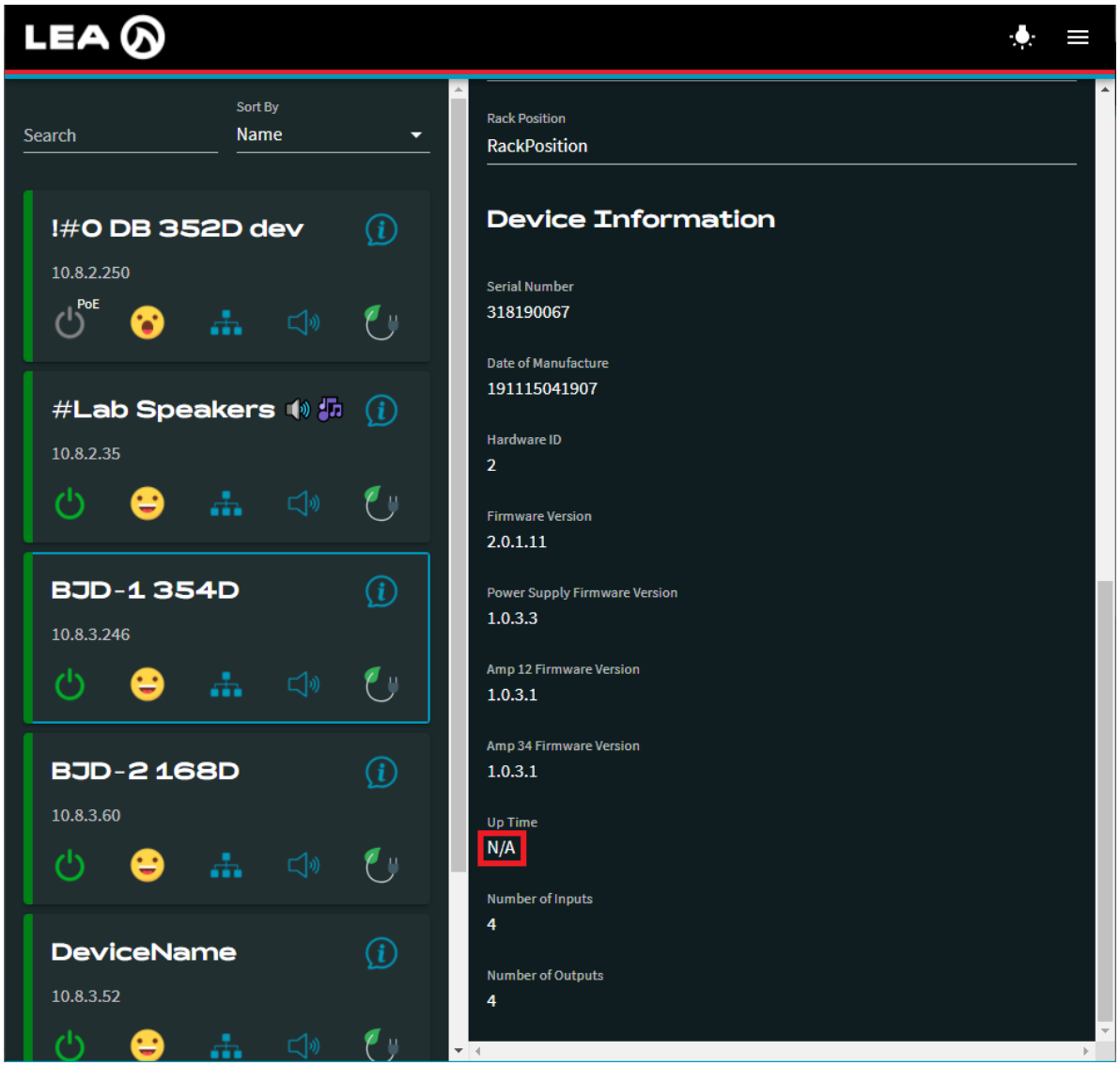
- Response: /amp/deviceInfo/amp34firmwareVersion "1.0.3.1"\n
- The command asked for the amplifiers amp 34 firmware version and got the response 1.0.3.1



Up Time

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/upTime
Values: any text up to 64 characters
Example: get /amp/deviceInfo/upTime\n

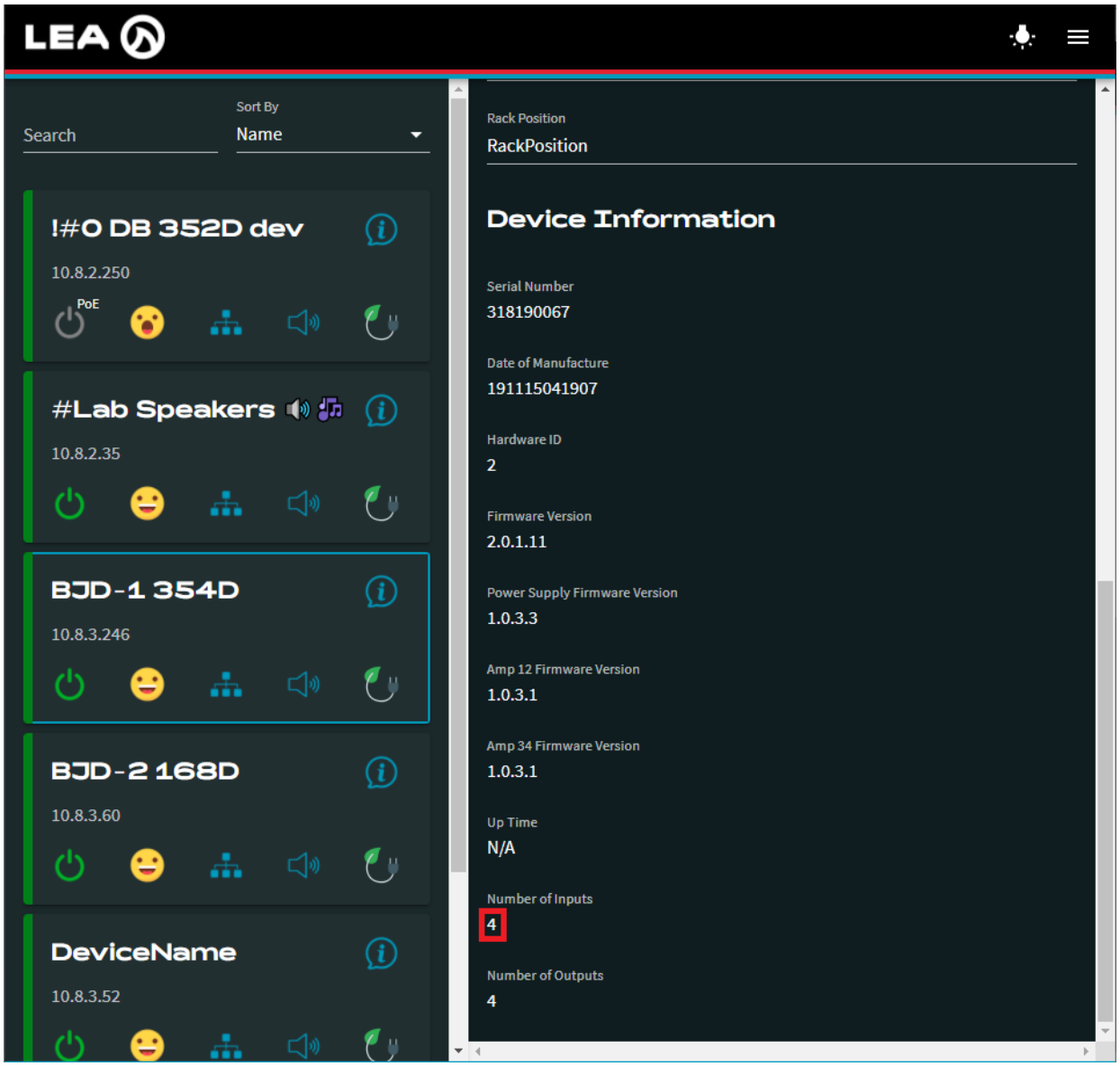
- Response: /amp/deviceInfo/upTime "N/A"\n
- The command asked for the amplifiers up time and got the response N/A



Number of Inputs

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/numInputs
Values: any text up to 64 characters
Example: get /amp/deviceInfo/numInputs\n

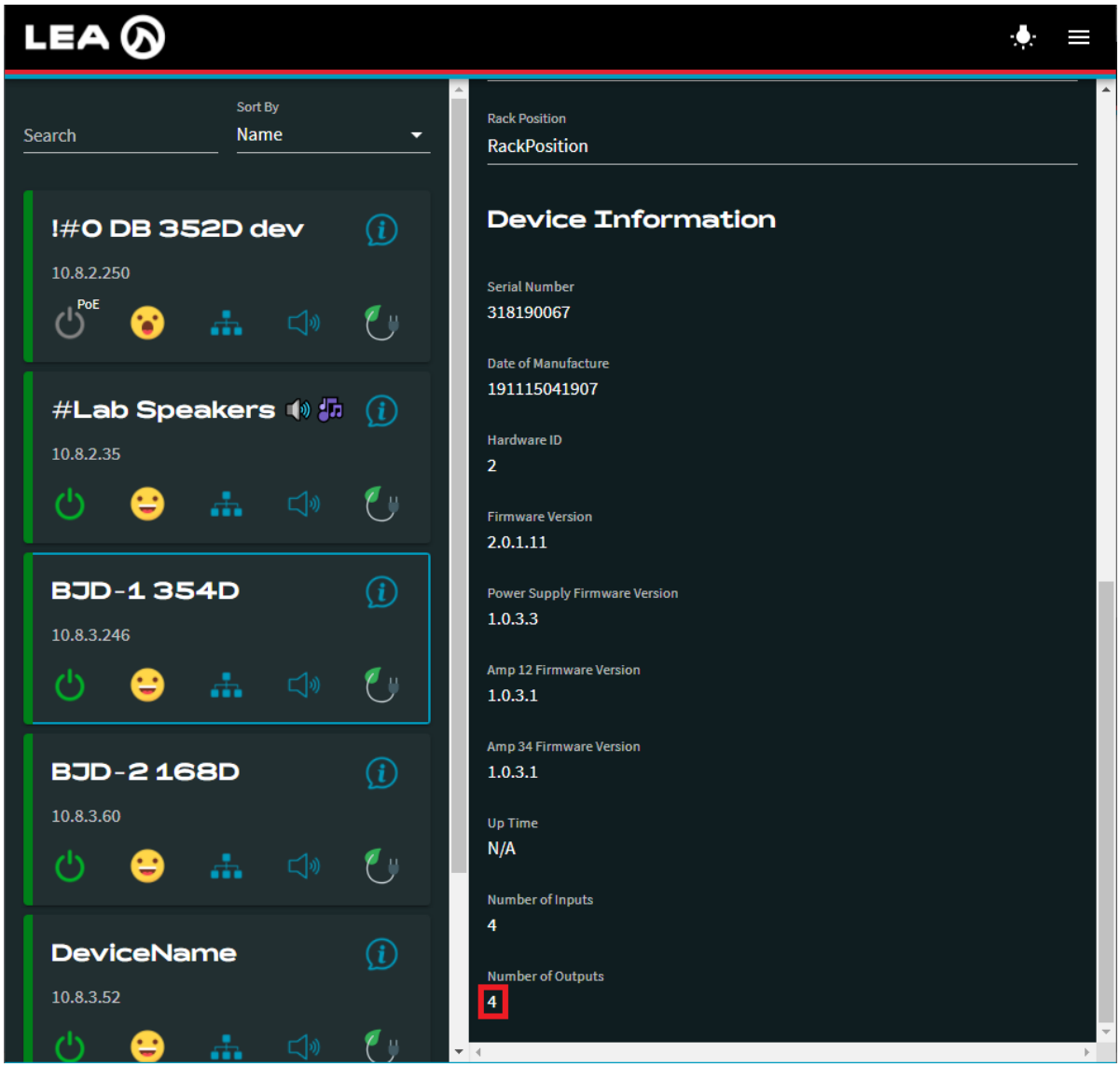
- Response: /amp/deviceInfo/numInputs 4.0\n
- The command asked for the amplifiers number of inputs and got the response 4



Number of Outputs

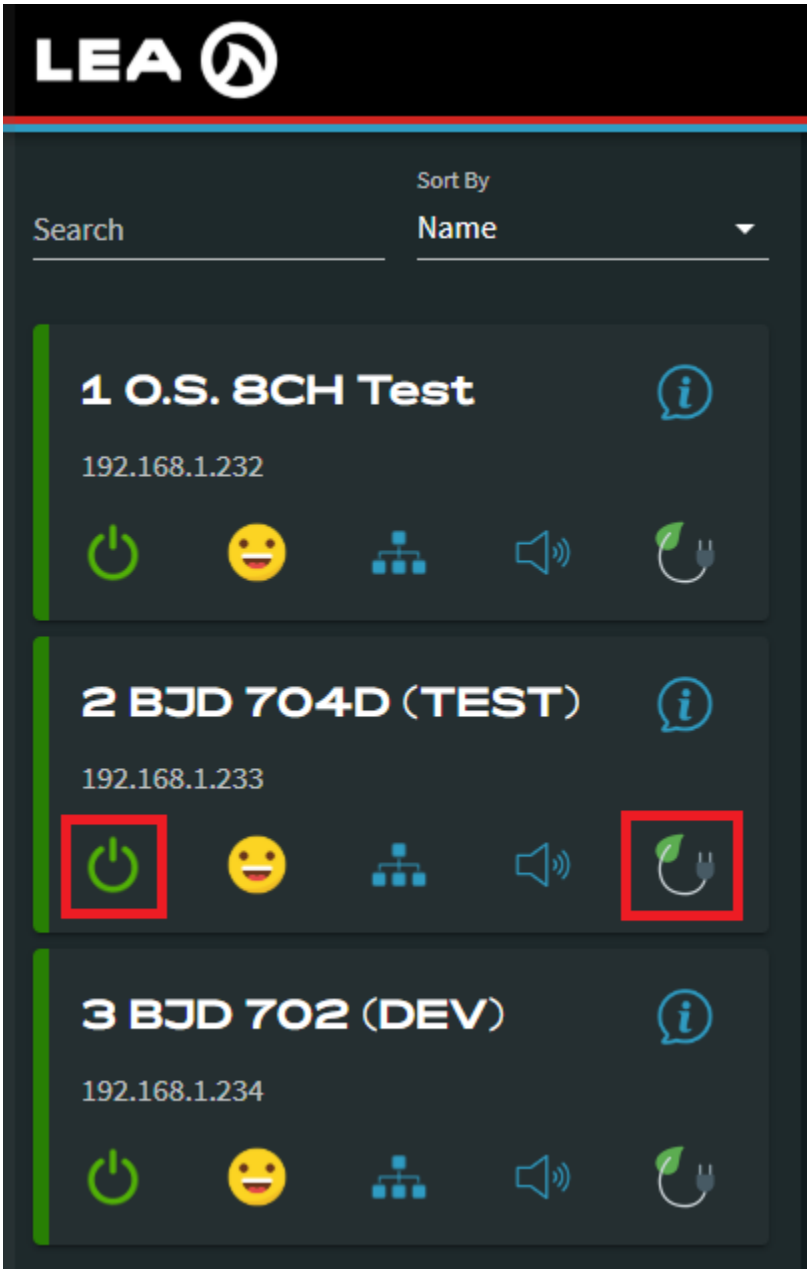
Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/deviceInfo/numOutputs
Values: any text up to 64 characters
Example: get /amp/deviceInfo/numOutputs\n

- Response: /amp/deviceInfo/numOutputs 4.0\n
- The command asked for the amplifiers number of outputs and got the response 4



Amplifier Power Supply

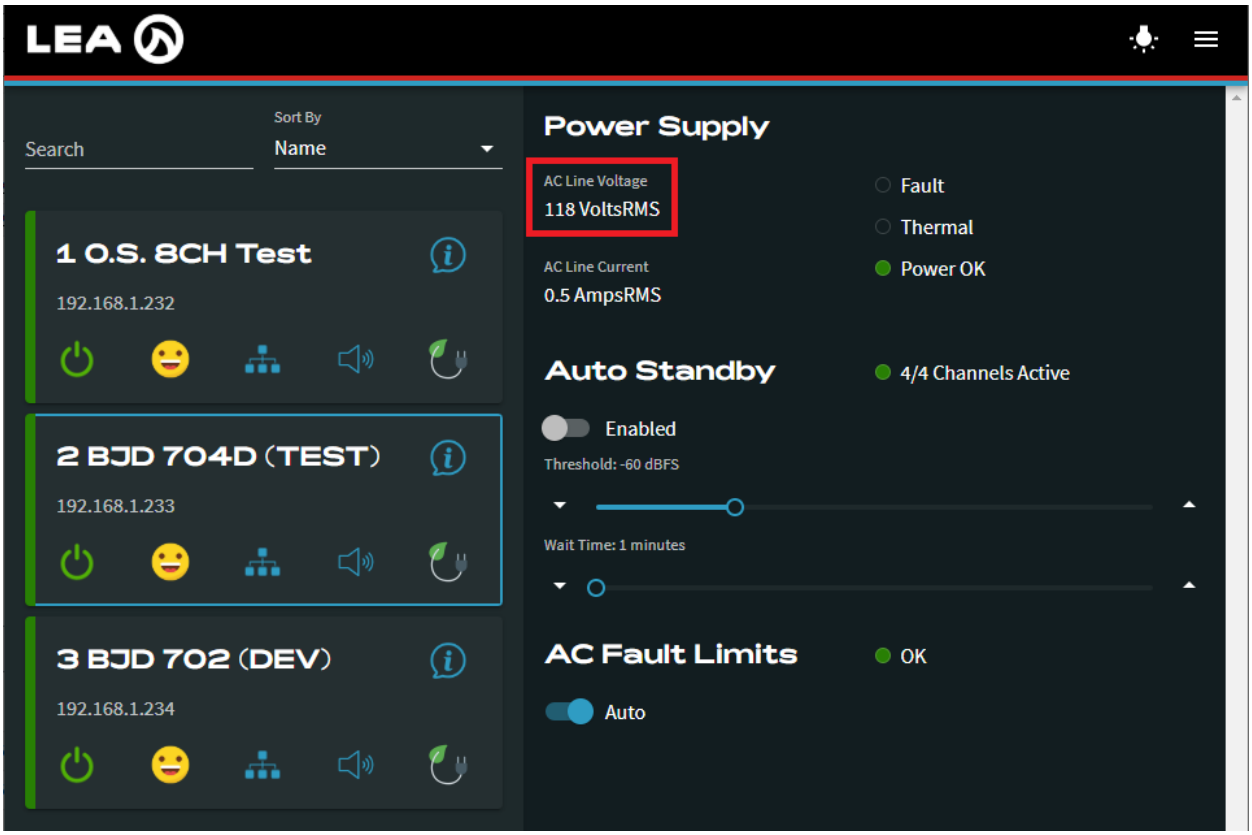
Click on the power supply or green power menu buttons to view the relevant pages for this API section



AC Line Voltage RMS

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/powerSupply/acLineVoltage
Values: 0.0 through 300.0 volts
Example: get /amp/powerSupply/acLineVoltage\n

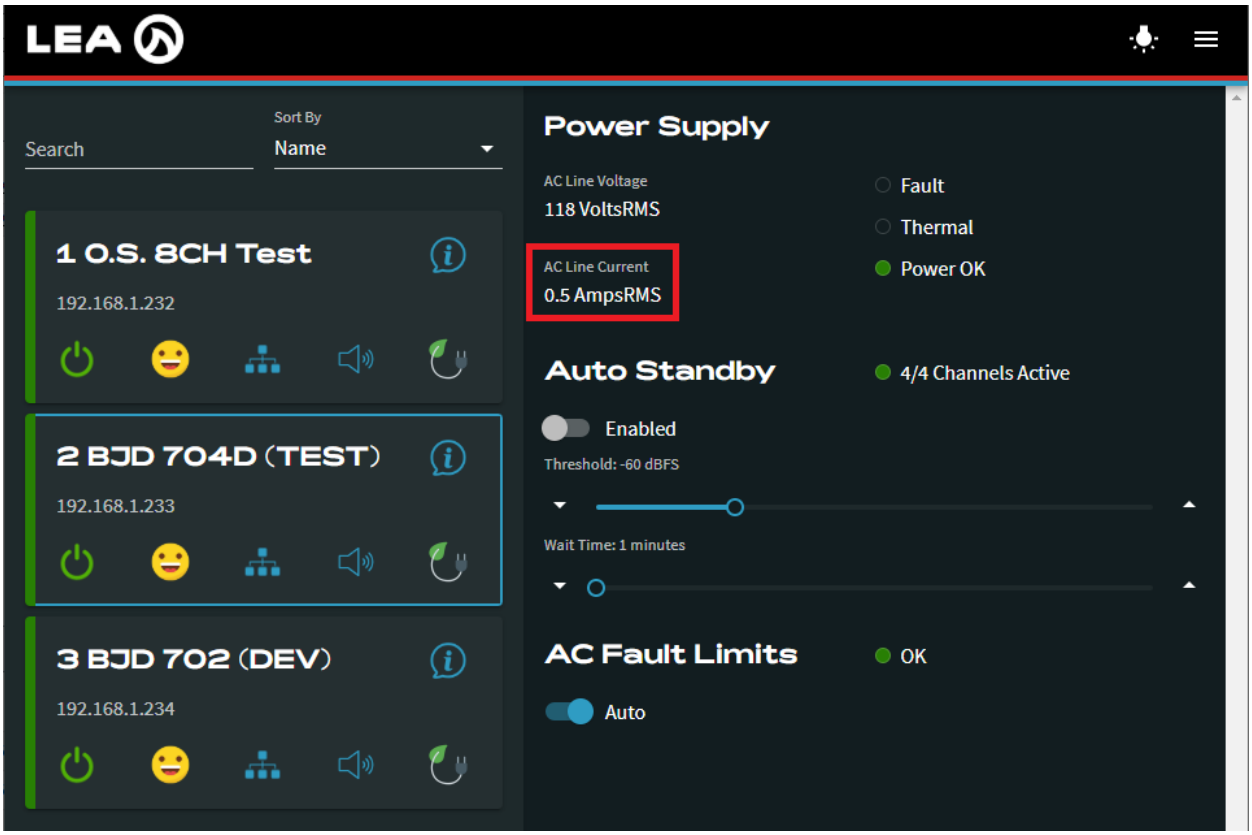
- Response: /amp/powerSupply/acLineVoltage 118.0\n
- The command asked for the amplifier AC Line Voltage RMS and got the response 118.0 V



AC Line Current RMS

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/powerSupply/acLineCurrent
Values: 0.0 through 100.0 amps
Example: get /amp/powerSupply/acLineCurrent\n

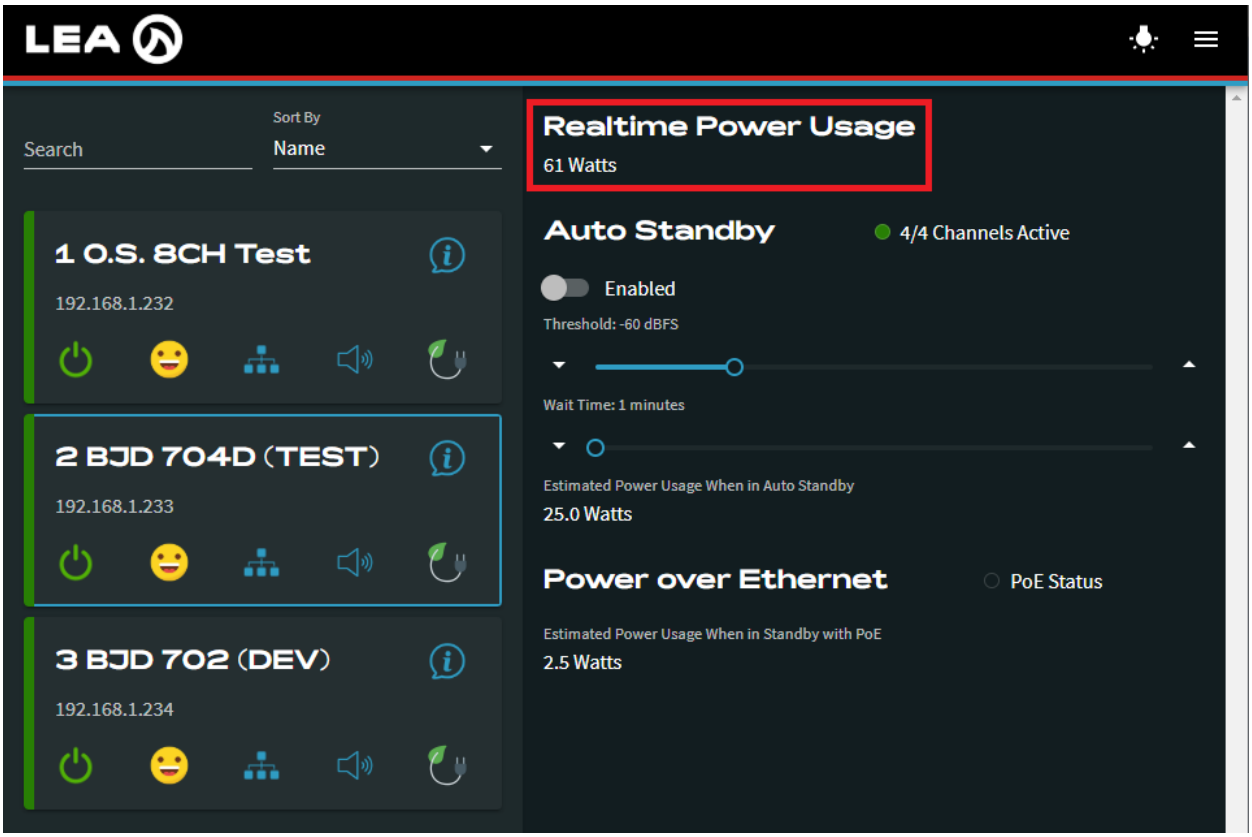
- Response: /amp/powerSupply/acLineCurrent 0.5\n
- The command asked for the amplifier AC Line Current RMS and got the response 0.5 amps



AC Line Power Draw

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/powerSupply/acLineWatts
Values: 0.0 through 5000.0
Example: get /amp/powerSupply/acLineWatts\n

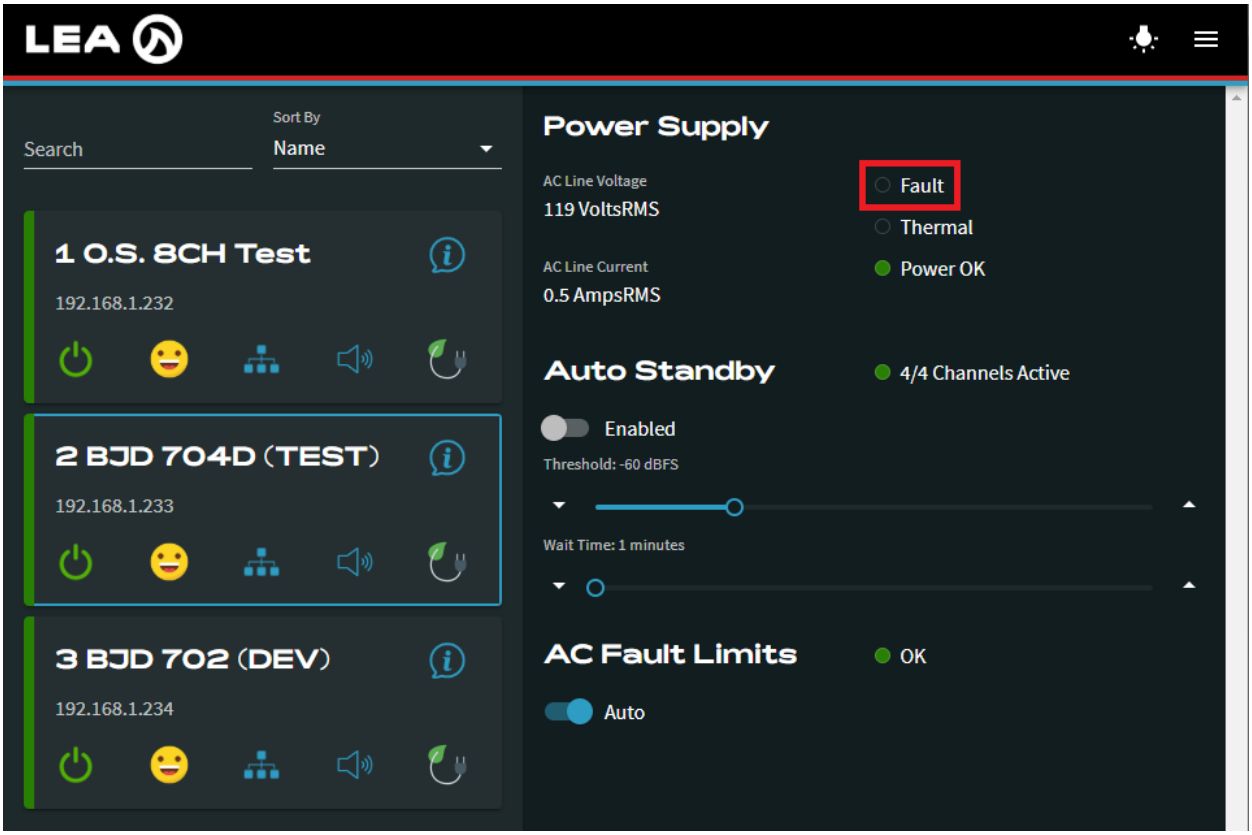
- Response: /amp/powerSupply/acLineWatts 61\n
- The command asked for the amplifier ac line watts and got the response 61 watts



Power Supply Fault Status

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/powerSupply/fault
Values: "true", "false"
Example: get /amp/powerSupply/fault\n

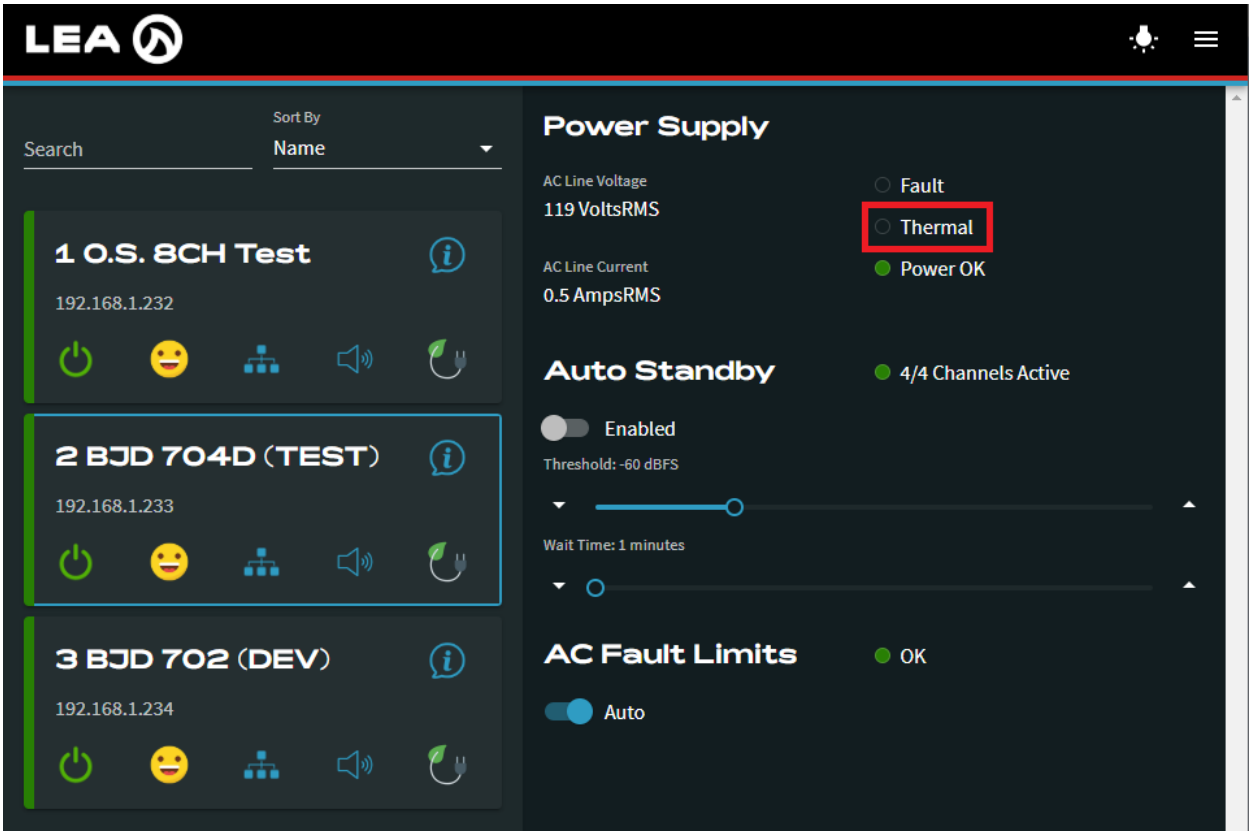
- Response: /amp/powerSupply/fault false\n
- The command asked for the power supply fault status and got the response False, meaning there is no power supply fault



Power Supply Thermal Protection Status

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/powerSupply/thermal
Values: "true", "false"
Example: get /amp/powerSupply/thermal\n

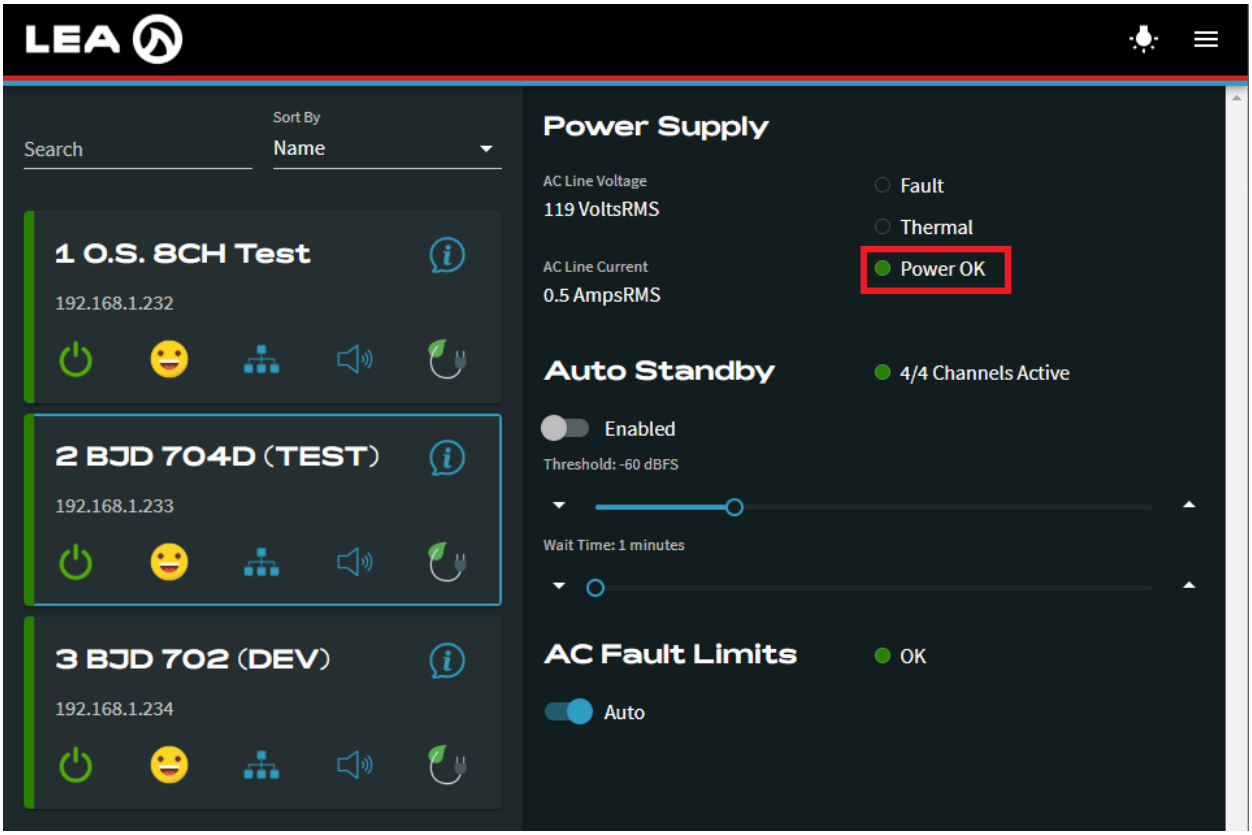
- Response: /amp/powerSupply/thermal false\n
- The command asked for the power supply thermal protection status and got the response False, meaning there is no thermal protection active, and the amplifier is operating at a safe temperature



Power Supply Power OK Status

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/powerSupply/powerOk
Values: "true", "false"
Example: get /amp/powerSupply/powerOk\n

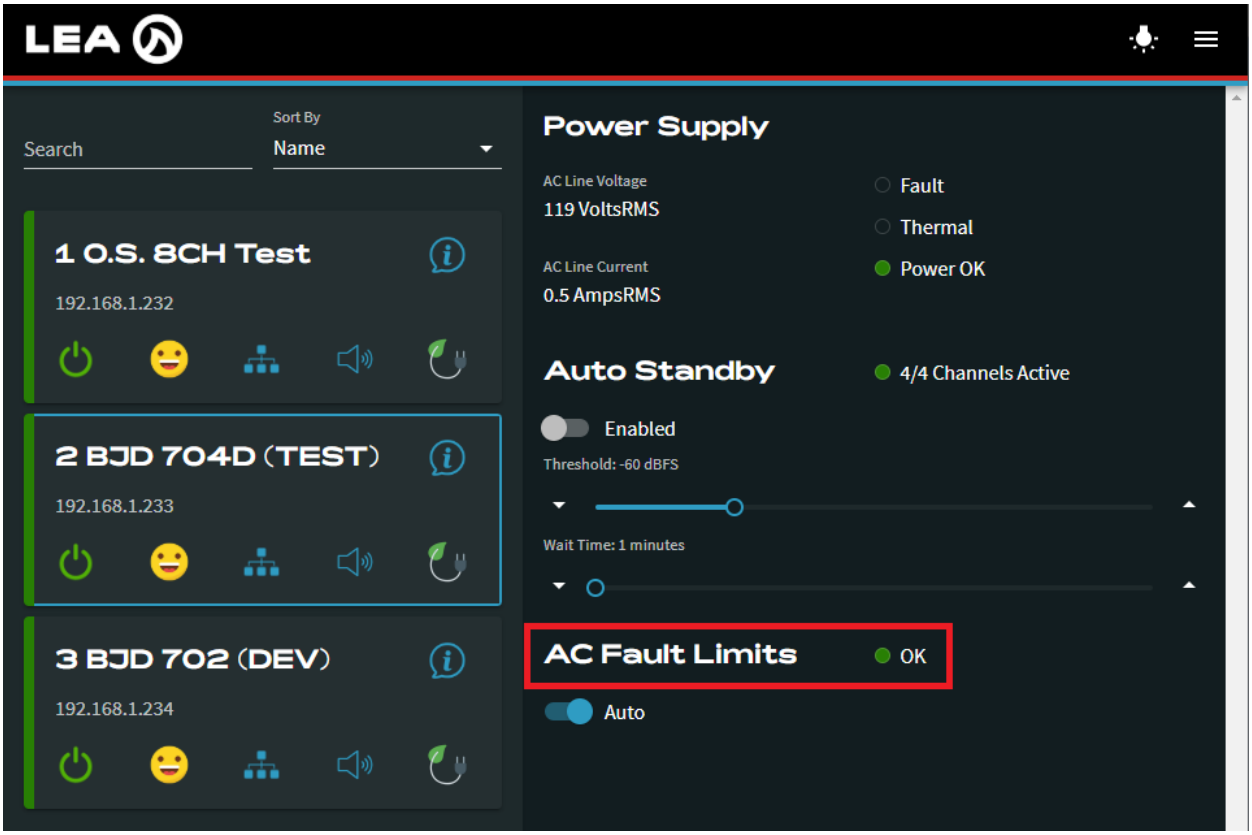
- Response: /amp/powerSupply/powerOk true\n
- The command asked for the power supply power ok status and got the response True, meaning that the power is Ok



Power Supply AC Line Voltage Ok

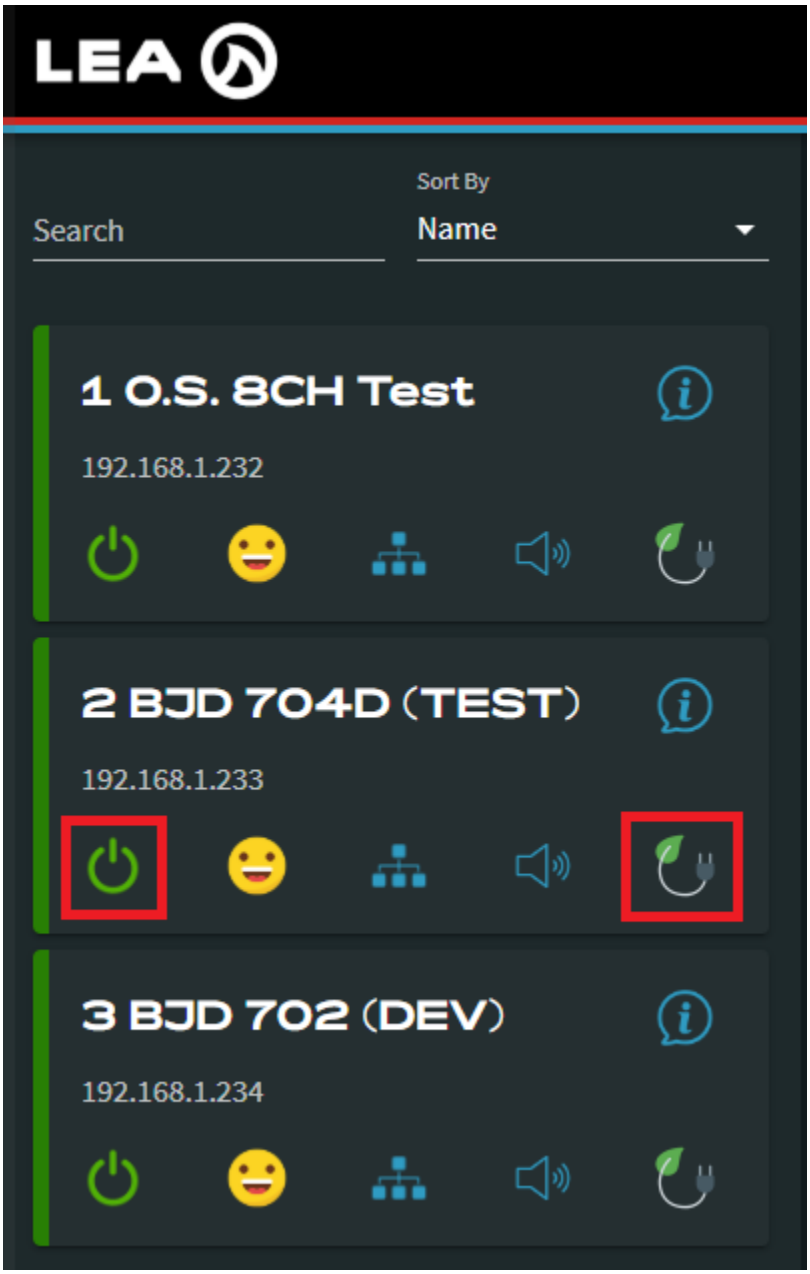
Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/powerSupply/lineWarning
Values: "true", "false"
Example: get /amp/powerSupply/lineWarning\n

- Response: /amp/powerSupply/lineWarning true\n
- The command asked for the power supply AC line voltage ok status and got the response True, meaning that it is Ok



Auto Standby

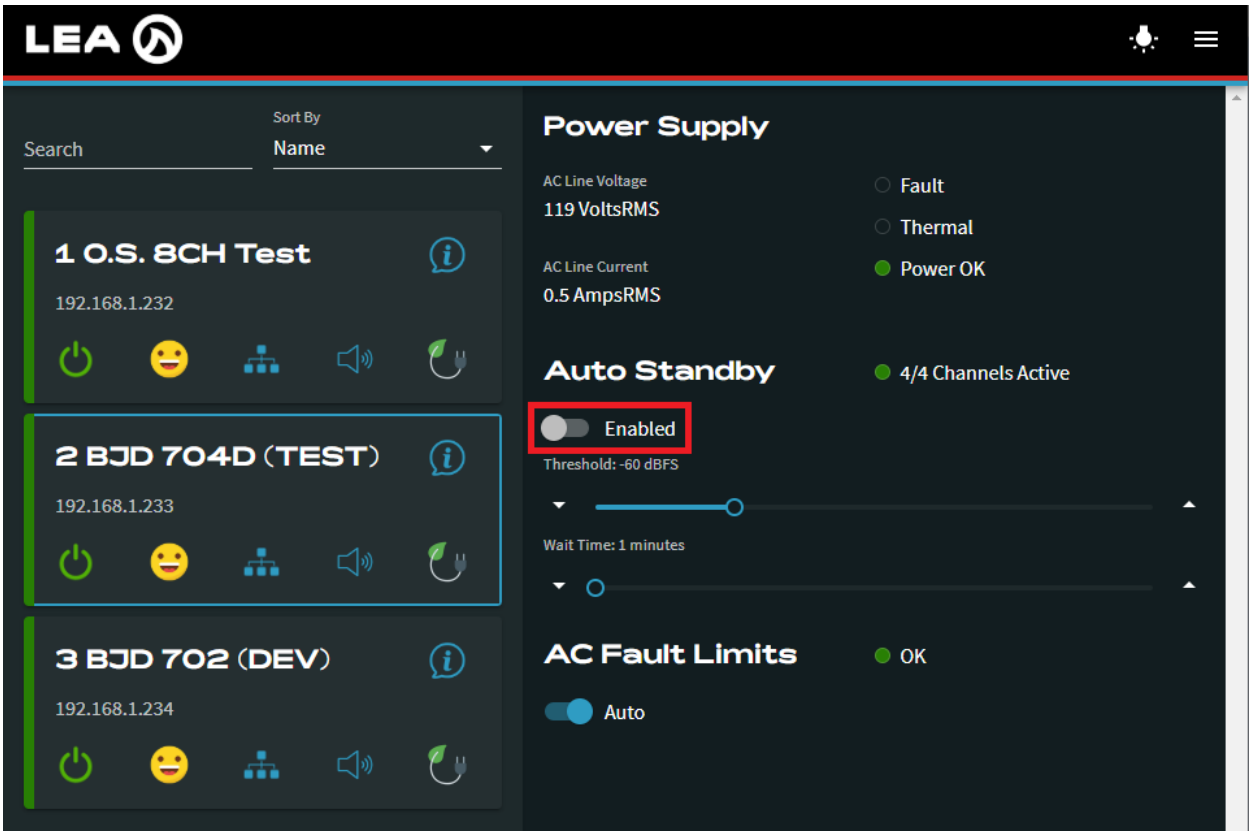
Click on the power supply or green power menu buttons to view the relevant pages for this API section



Auto Standby Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/autoStandby/enable
Values: "true", "false"
Example: set /amp/autoStandby/enable "true"\n

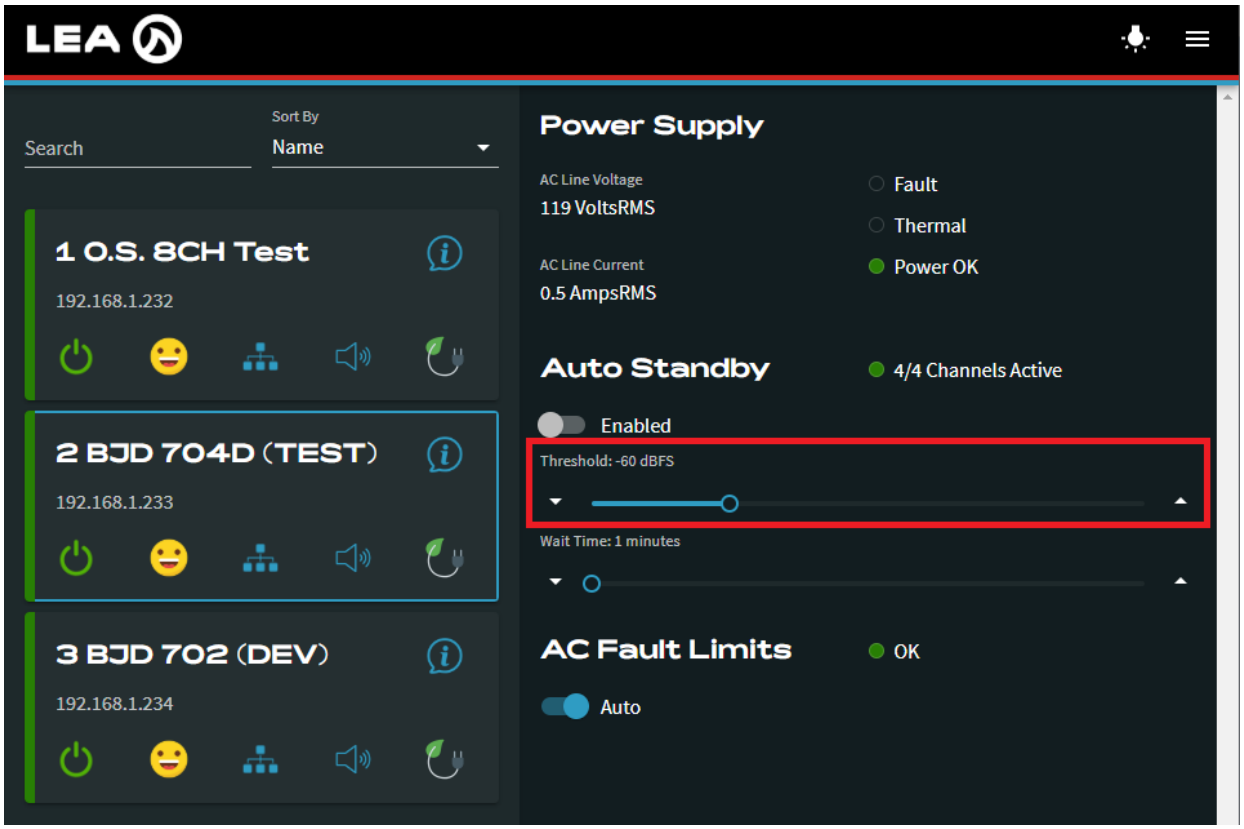
- Response: OK\n
- The command set the auto standby setting to enabled



Auto Standby Threshold

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/autoStandby/threshold
Values: -80.0 through 0.0
Example: set /amp/autoStandby/threshold -60.0\n

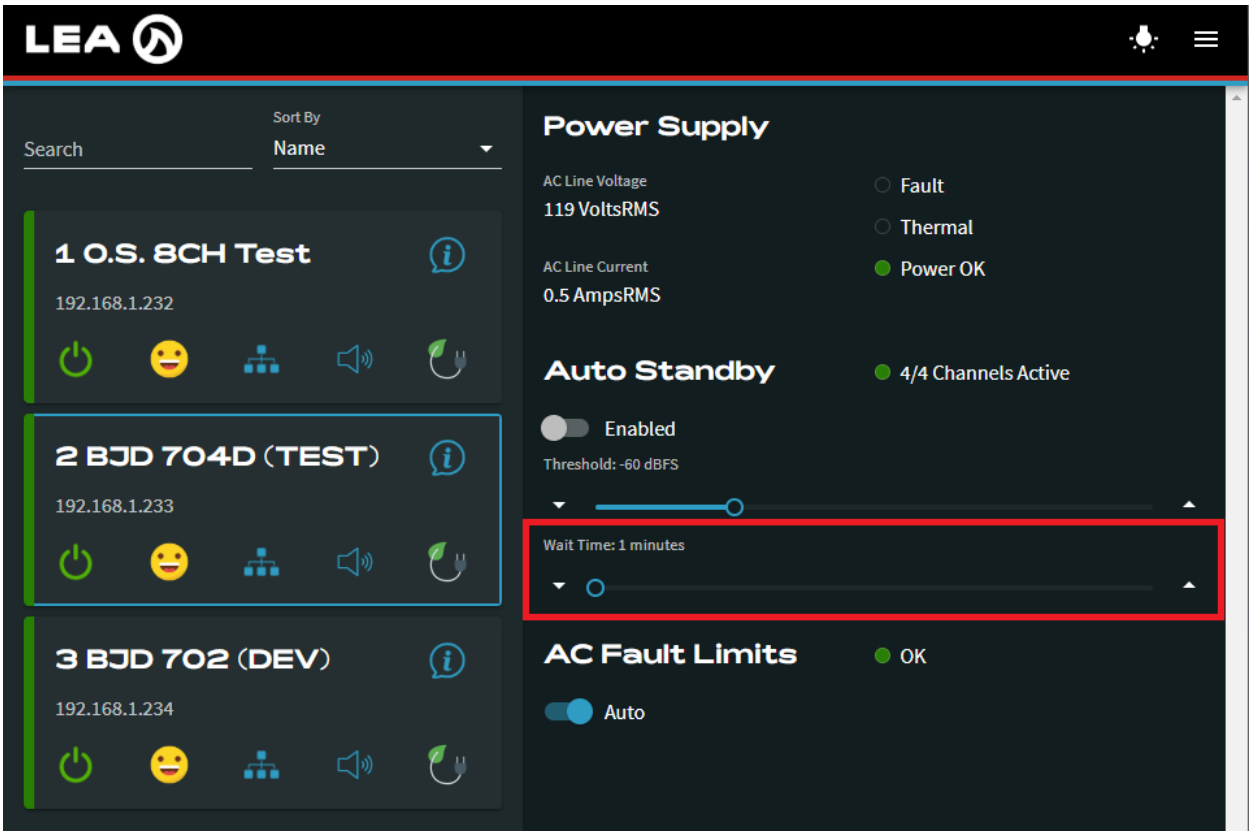
- Response: OK\n
- The command set the auto standby threshold to -60.0 dBFS



Auto Standby Wait Time

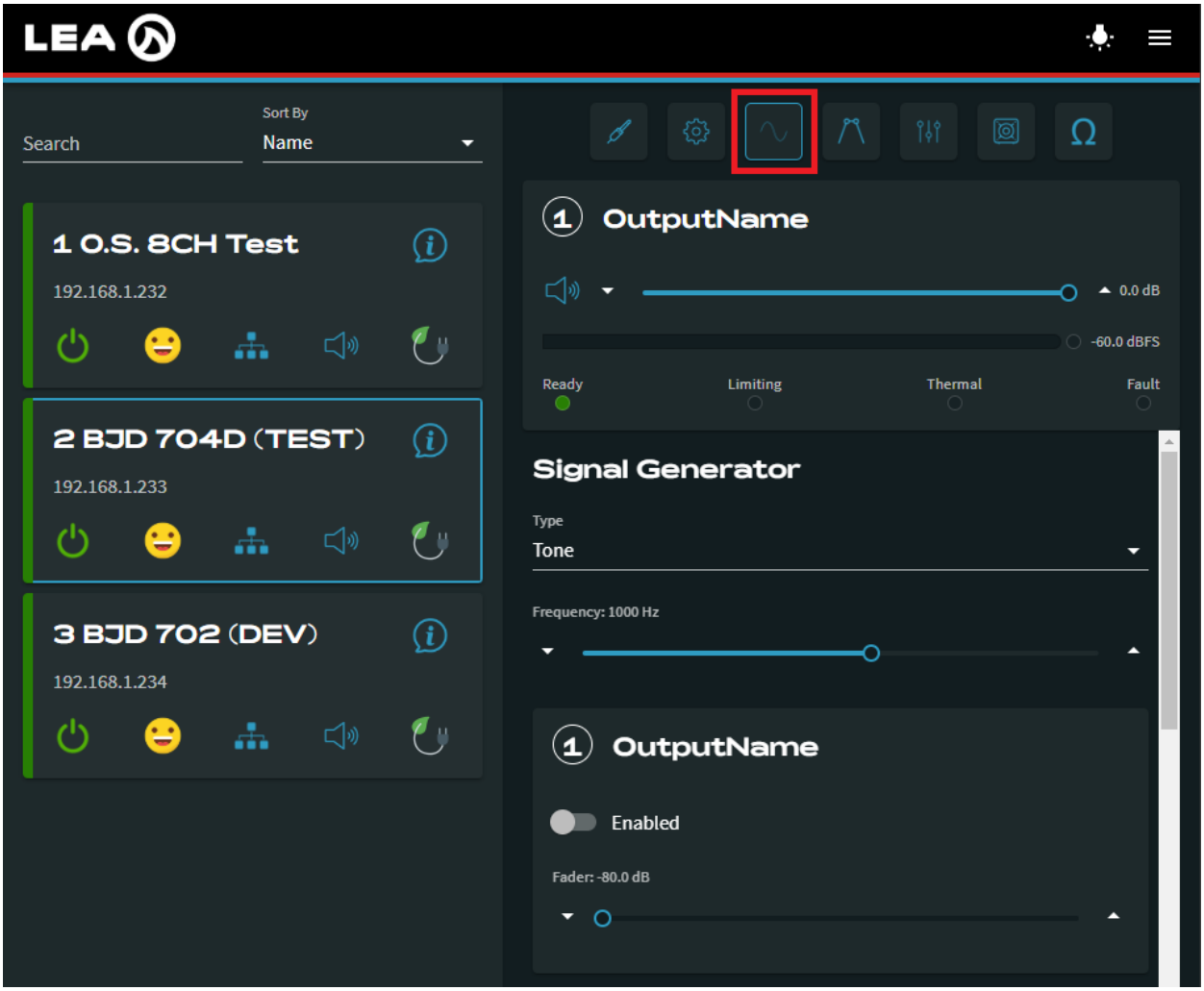
Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/autoStandby/timeToWait
Values: 1 through 240
Example: set /amp/autoStandby/timeToWait 1\n

- Response: OK\n
- The command set the auto standby wait time to 1 minute



Signal Generator

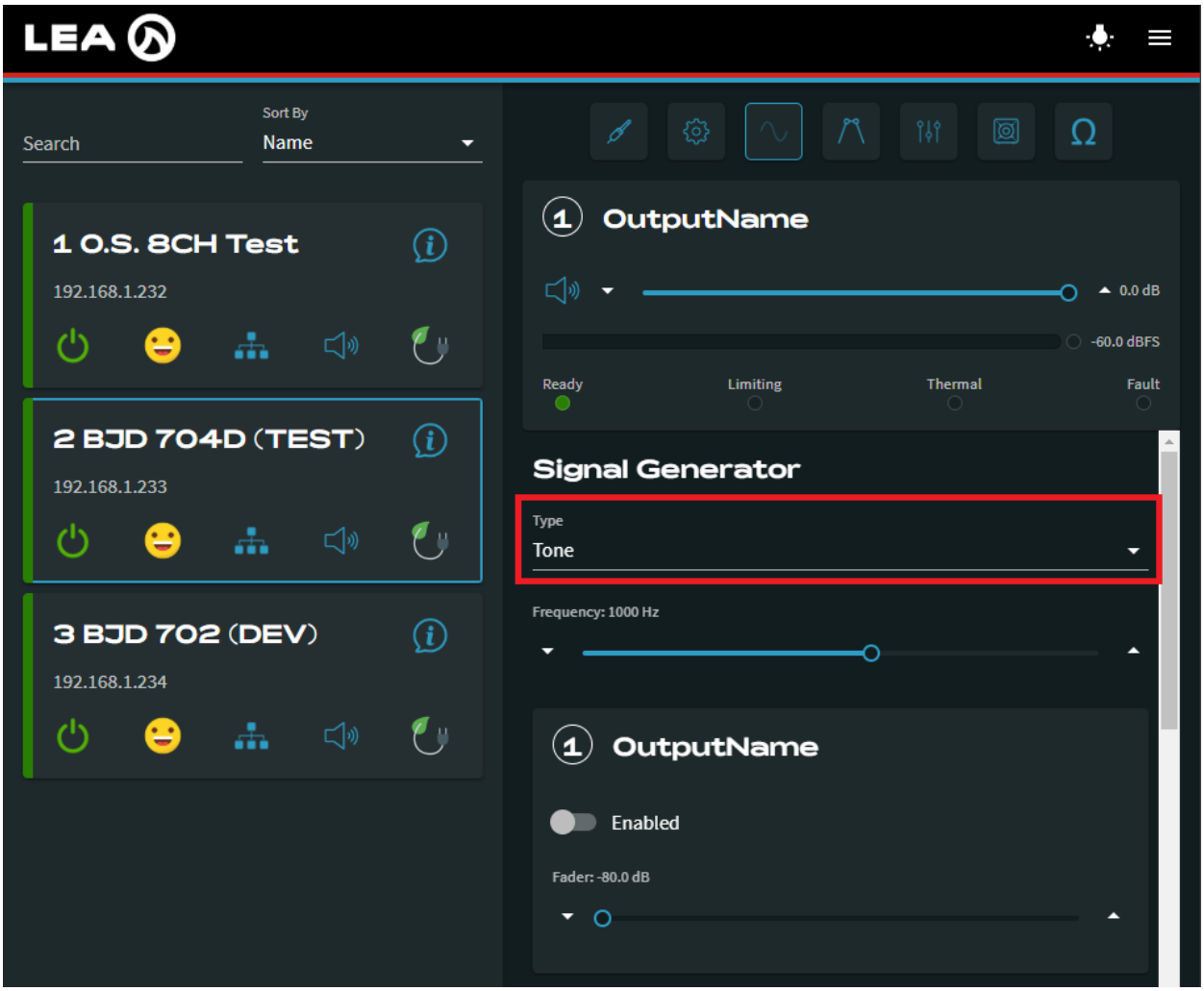
Click on this button to navigate to the signal generator page relevant to this API section



Signal Generator Type

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/signalGenerator/type
Values: "Pink Noise", "White Noise", "Tone"
Example: set /amp/signalGenerator/type "Tone"\n

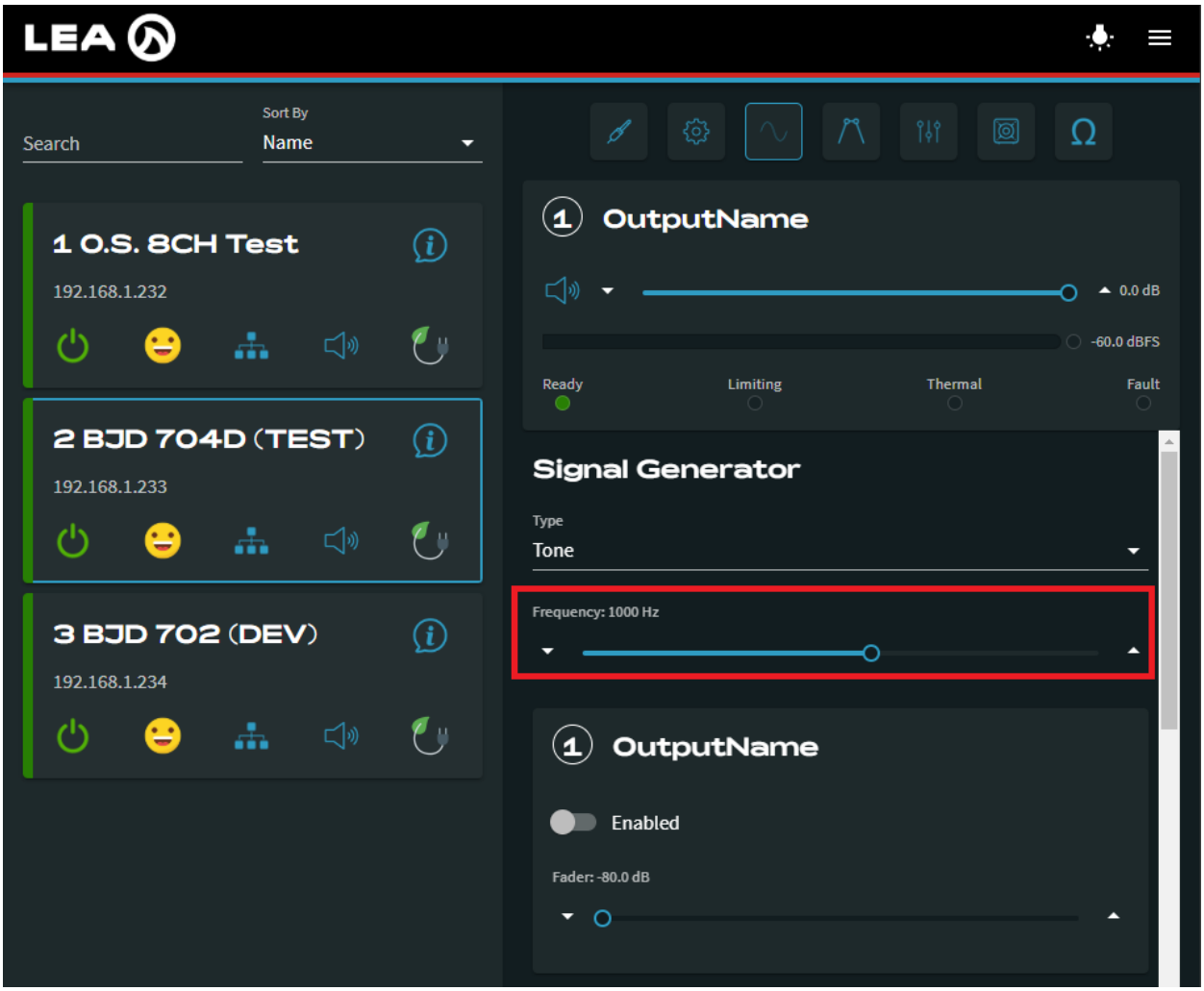
- Response: OK\n
- The command set the signal generator type to sine wave tone



Signal Generator Tone Frequency

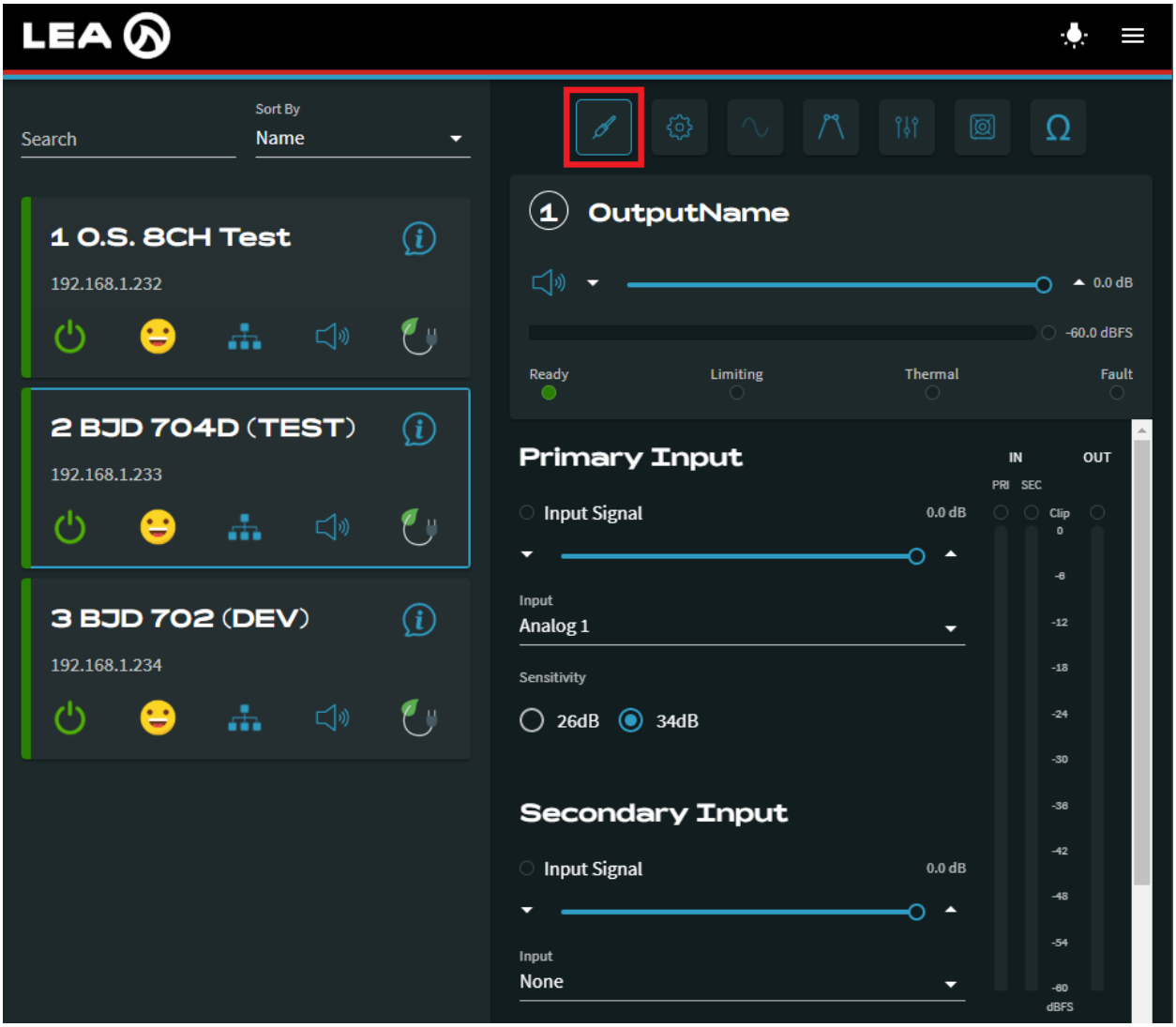
Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/signalGenerator/frequency
Values: 20 through 20000
Example: set /amp/signalGenerator/frequency 1000\n

- Response: OK\n
- The command set the signal generator sine wave tone frequency to 1000 Hz



Amplifier Analog Inputs

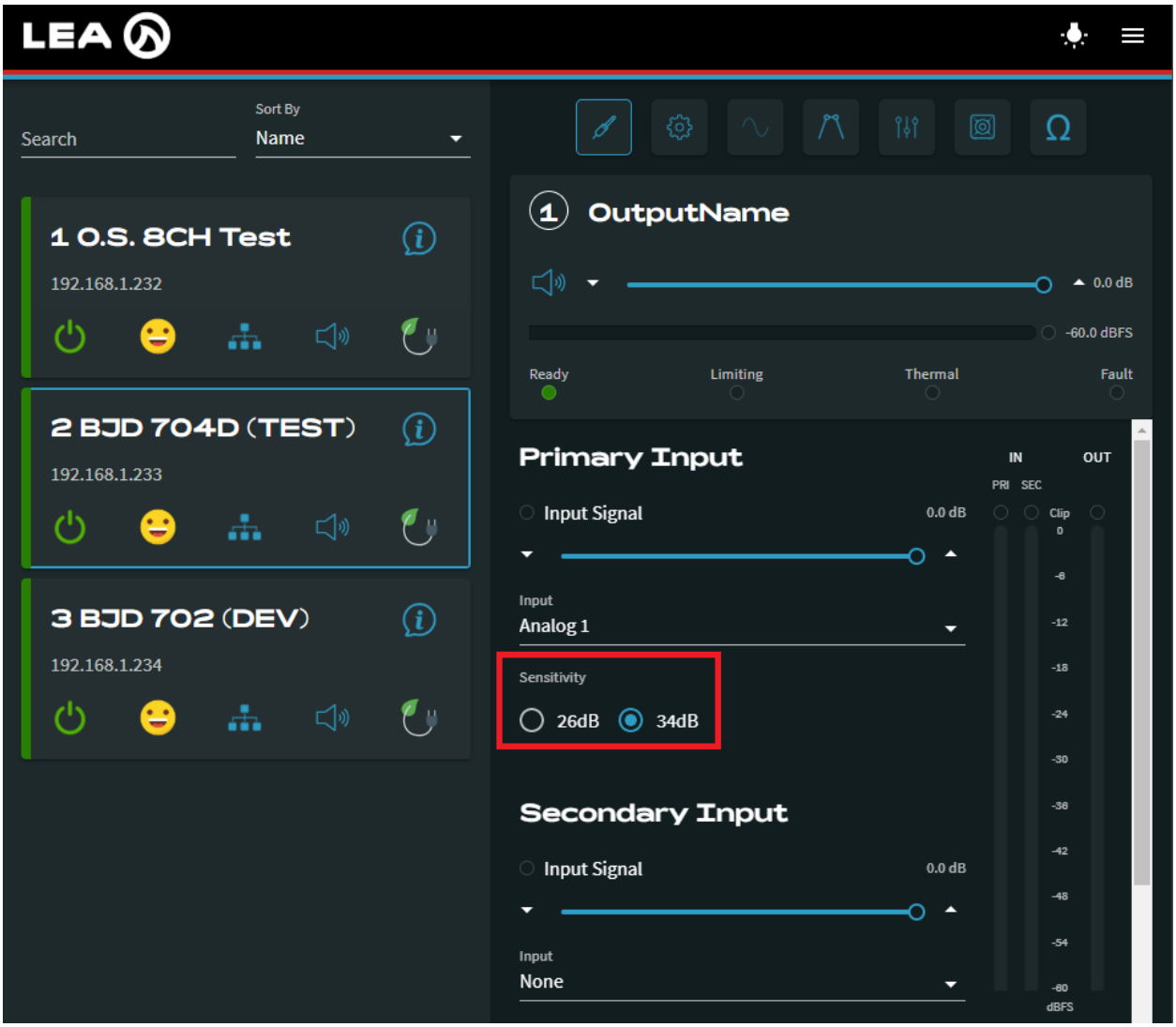
Click on this button to navigate to the Input section of the DSP relevant to this API section



Analog Input Sensitivity

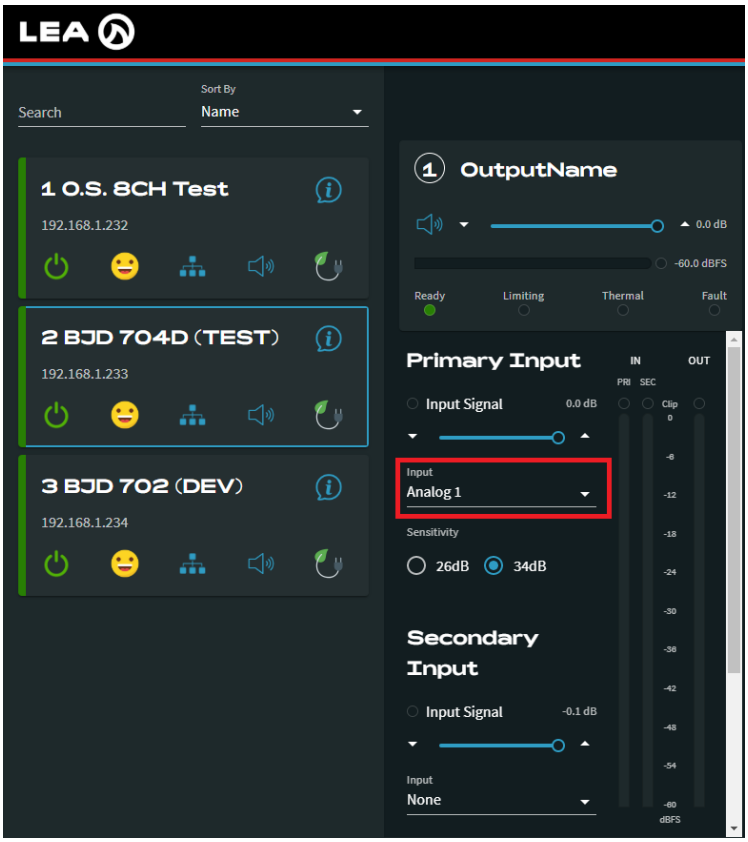
Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/inputs/analog/#/sensitivity
Values: "26dB", "34dB"
Example: set /amp/inputs/analog/1/sensitivity "34dB"\n

- Response: OK\n
- The command set the sensitivity of the Analog 1 input to 34dB



Primary Input Source

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/primary
Values:
"Analog 1"
"Analog 2"
"Analog 1+2"
"Analog 3"
"Analog 4"
"Analog 3+4"
"Analog 5"
"Analog 6"
"Analog 5+6"
"Analog 7"
"Analog 8"
"Analog 7+8"
"Dante 1"
"Dante 2"
"Dante 1+2"
"Dante 3"
"Dante 4"
"Dante 3+4"
"Dante 5"
"Dante 6"
"Dante 5+6"
"Dante 7"
"Dante 8"
"Dante 7+8"



Example: set /amp/channels/1/inputSelector/primary "Analog 2"\n

- This will set the primary input source on Channel 1 to "Analog 2"

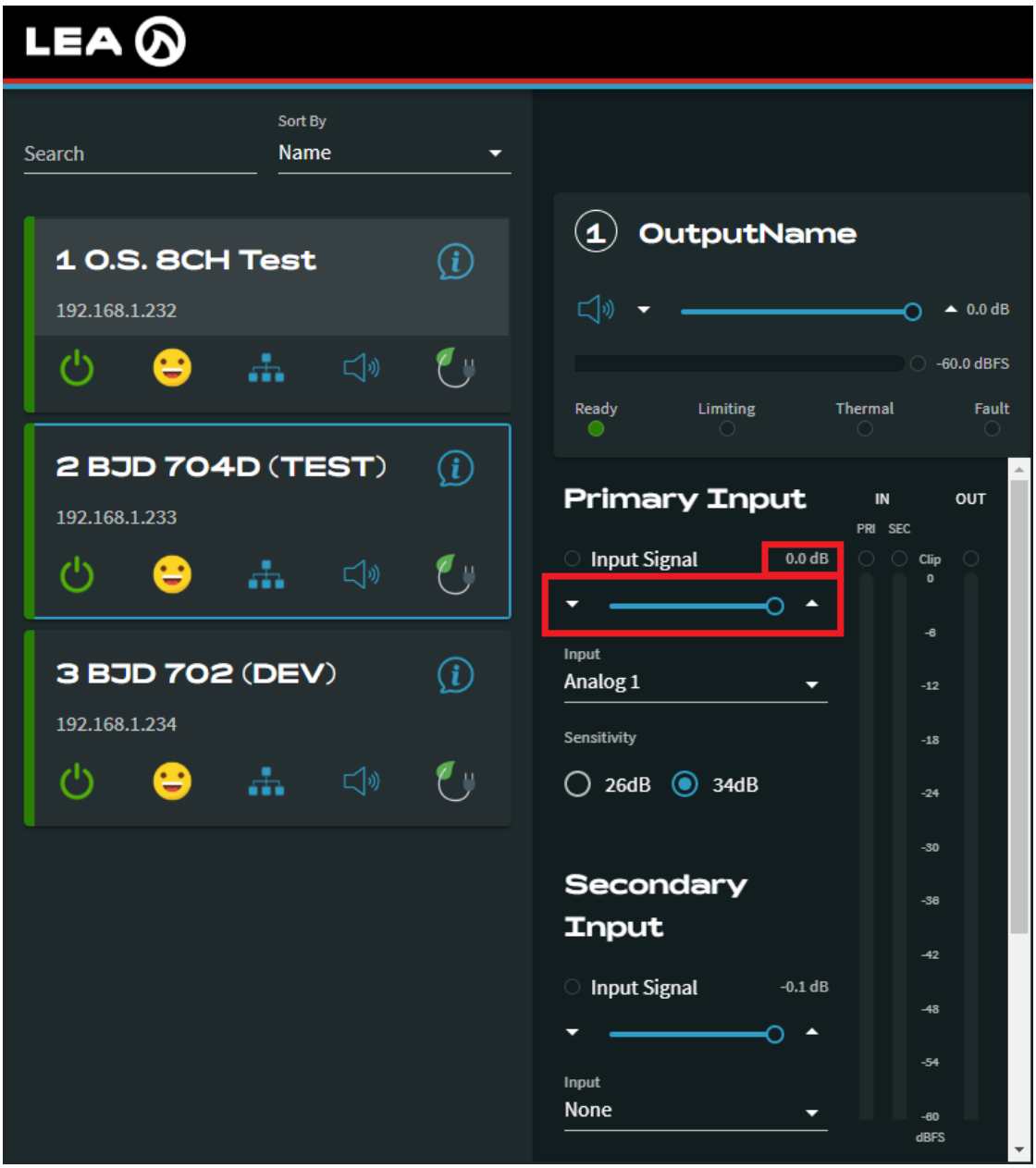
For the Network Connect Series, only Analog inputs are available, and only the number of channels that are on the amplifier, for example a 354 would not have Analog 5-8 available.

For the Dante Connect Series analog inputs are only available for the number of inputs on the amplifier, however all 8 Dante inputs are available on all Dante models.

Primary Gain Attenuation Fader

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/primaryFader
Values: Gain attenuation values between -80 and 0
Example: set /amp/channels/2/inputSelector/primaryFader -10.0\n

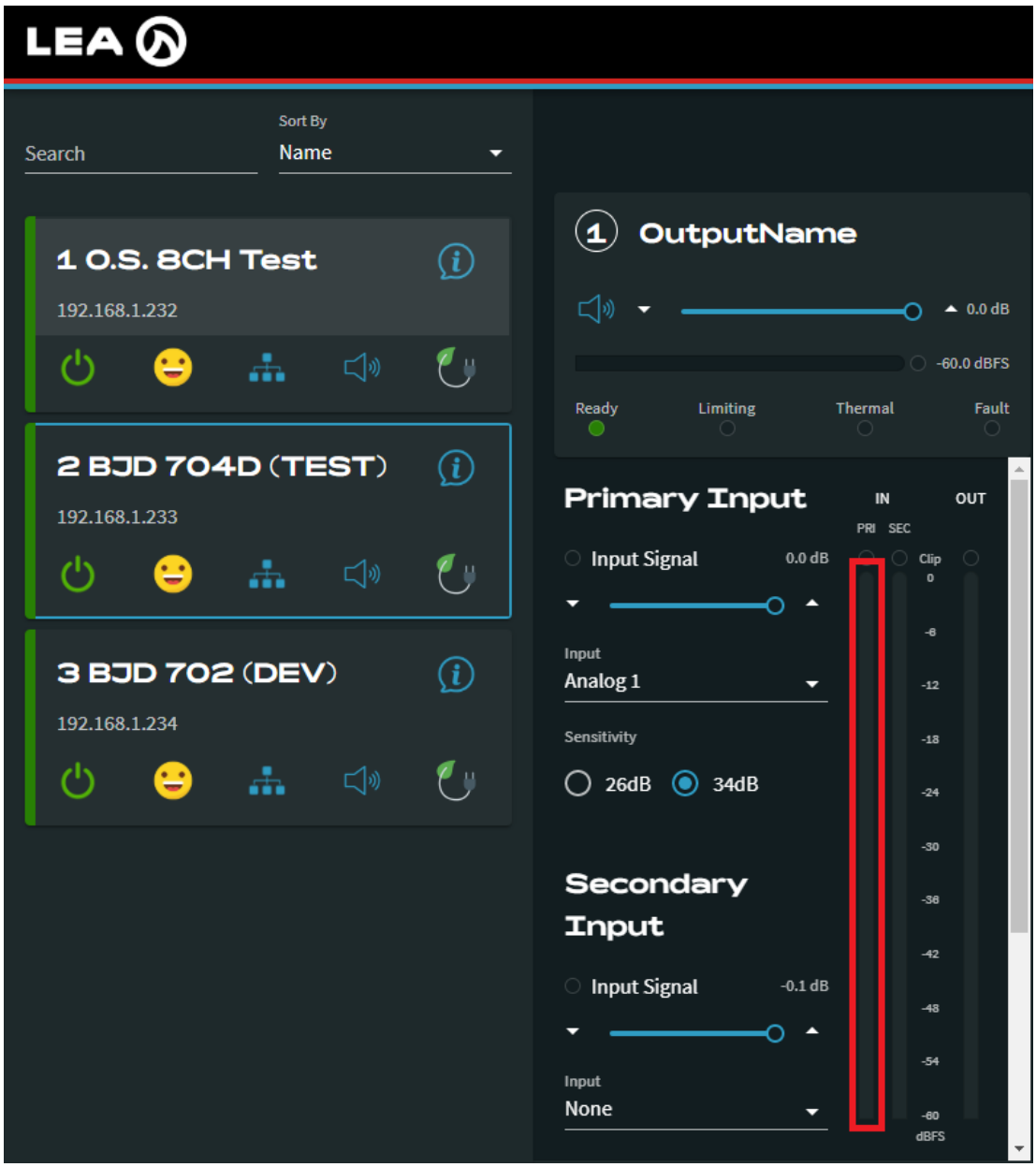
- This will set the Primary Input Level on Channel 2 to -10.0dB



Primary Input Meter

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/primaryLevel
Values: -80 through 0 dBFS
Example: subscribe /amp/channels/1/inputSelector/primaryLevel\n

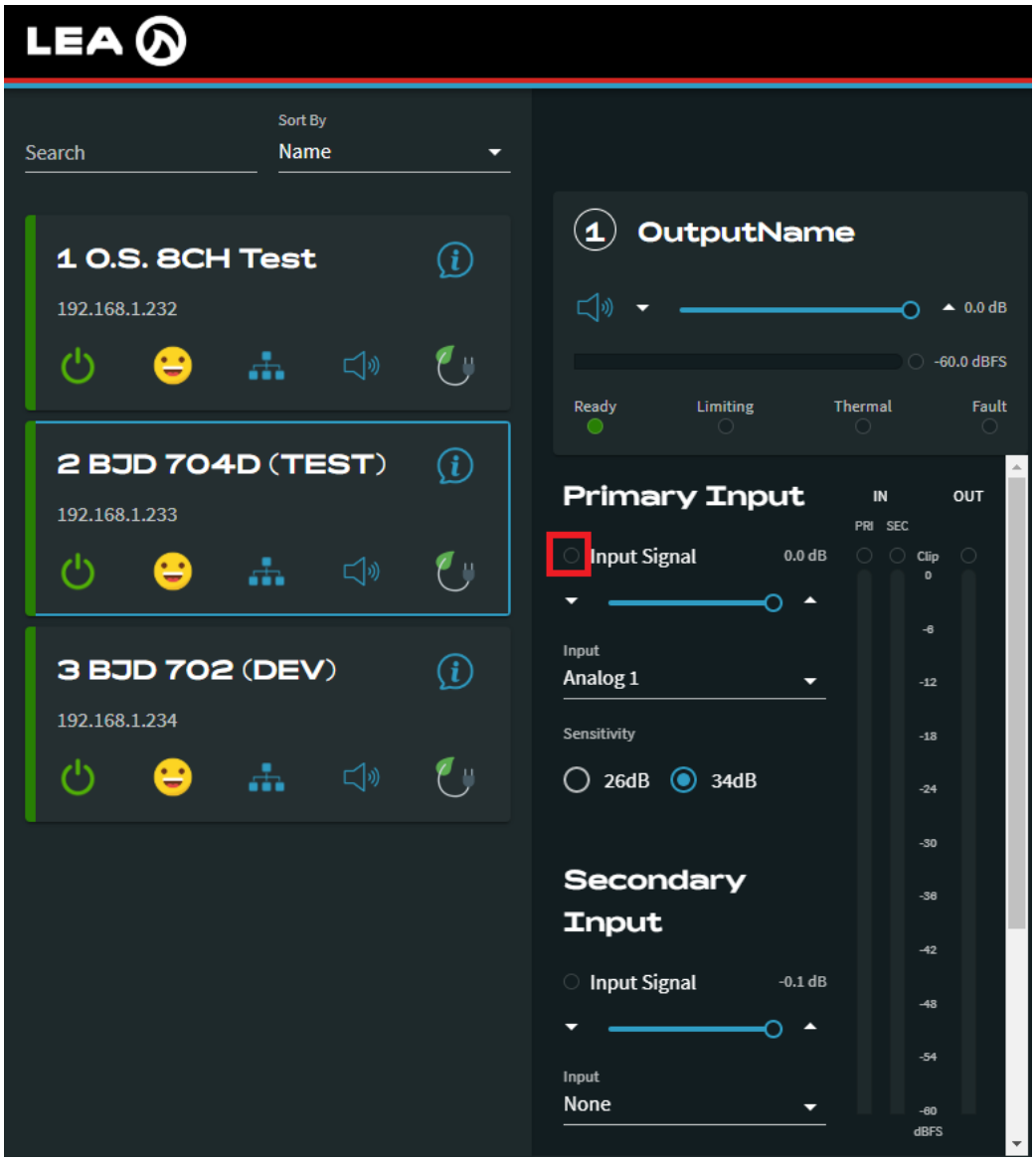
- This will subscribe to the primary input meter level for channel 1



Primary Input Signal Present

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/primarySignalDetect
Values: "true", "false"
Example: get /amp/channels/1/inputSelector/primarySignalDetect\n

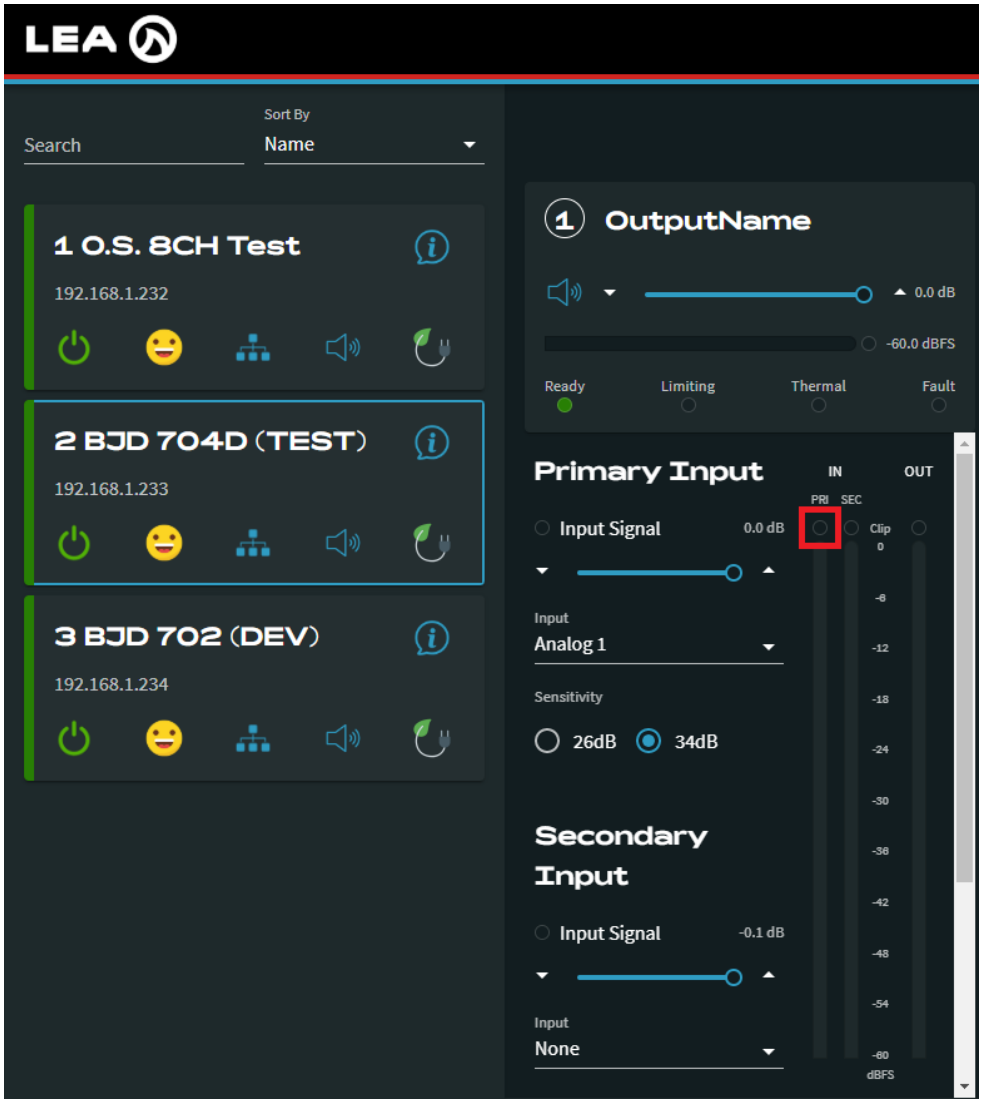
- Response: /amp/channels/1/inputSelector/primarySignalDetect true\n
- This command asked to get the primary input signal presence state on Channel 1 and the response back was "true" meaning that signal is present



Primary Input Signal Clip Indicator

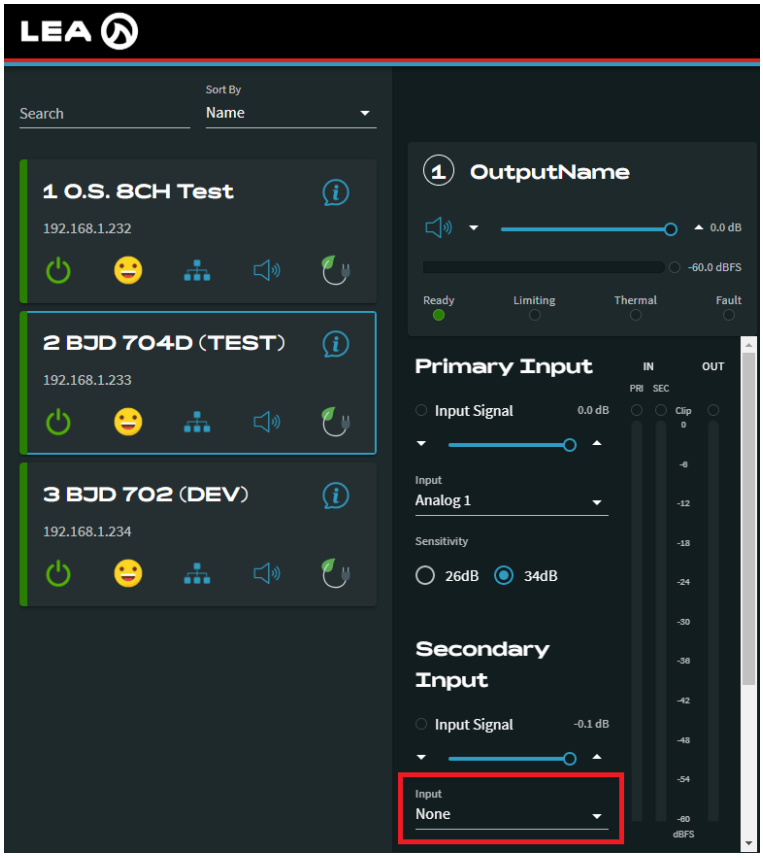
Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/primaryClip
Values: "true", "false"
Example: subscribe /amp/channels/1/inputSelector/primaryClip\n

- Response: /amp/channels/1/inputSelector/primaryClip false\n
- This command asked to subscribe to primary input signal clip indication state on Channel 1 and the response back was "false" meaning that signal is not clipped, but as this is a subscription and any changes here will continue to be reported



Secondary Input Source

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/secondary
Values:
"Analog 1"
"Analog 2"
"Analog 1+2"
"Analog 3"
"Analog 4"
"Analog 3+4"
"Analog 5"
"Analog 6"
"Analog 5+6"
"Analog 7"
"Analog 8"
"Analog 7+8"
"Dante 1"
"Dante 2"
"Dante 1+2"
"Dante 3"
"Dante 4"
"Dante 3+4"
"Dante 5"
"Dante 6"
"Dante 5+6"
"Dante 7"
"Dante 8"
"Dante 7+8"
"None"



Example: set /amp/channels/1/inputSelector/secondary "Analog 2"\n

- This will set the secondary input source on Channel 1 to "Analog 2"

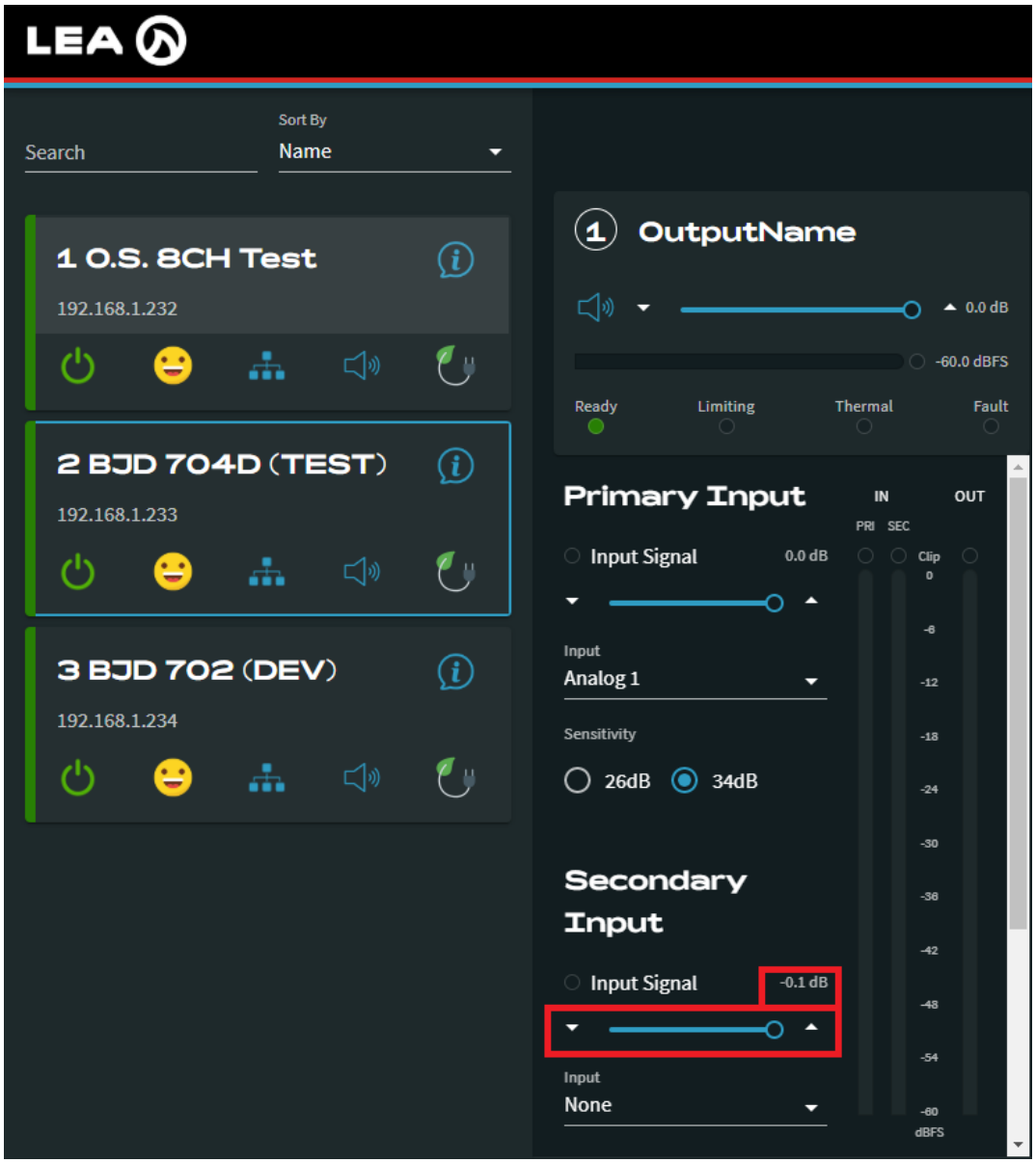
For the Network Connect Series, only Analog inputs are available, and only the number of channels that are on the amplifier, for example a 354 would not have Analog 5-8 available.

For the Dante Connect Series analog inputs are only available for the number of inputs on the amplifier, however all 8 Dante inputs are available on all Dante models.

Secondary Gain Attenuation Fader

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/secondaryFader
Values: Gain attenuation values between -80 and 0
Example: get /amp/channels/1/inputSelector/secondaryFader\n

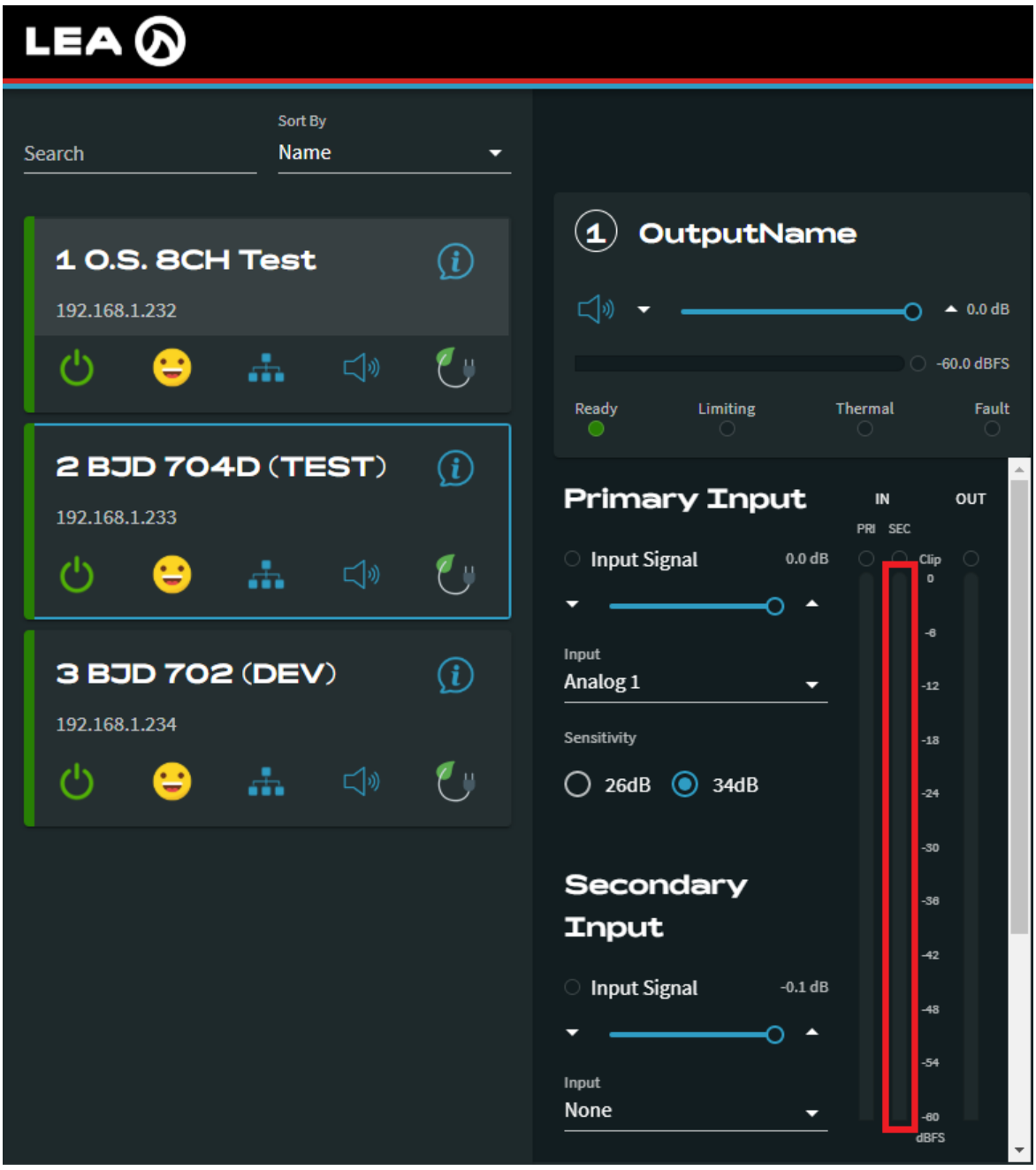
- Response: /amp/channels/1/inputSelector/secondaryFader -0.1\n
- This command asked for the secondary fader level on channel 1 and got -0.1dB as the response



Secondary Input Meter

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/secondaryLevel
Values: -80 through 0 dBFS
Example: subscribe /amp/channels/1/inputSelector/secondaryLevel\n

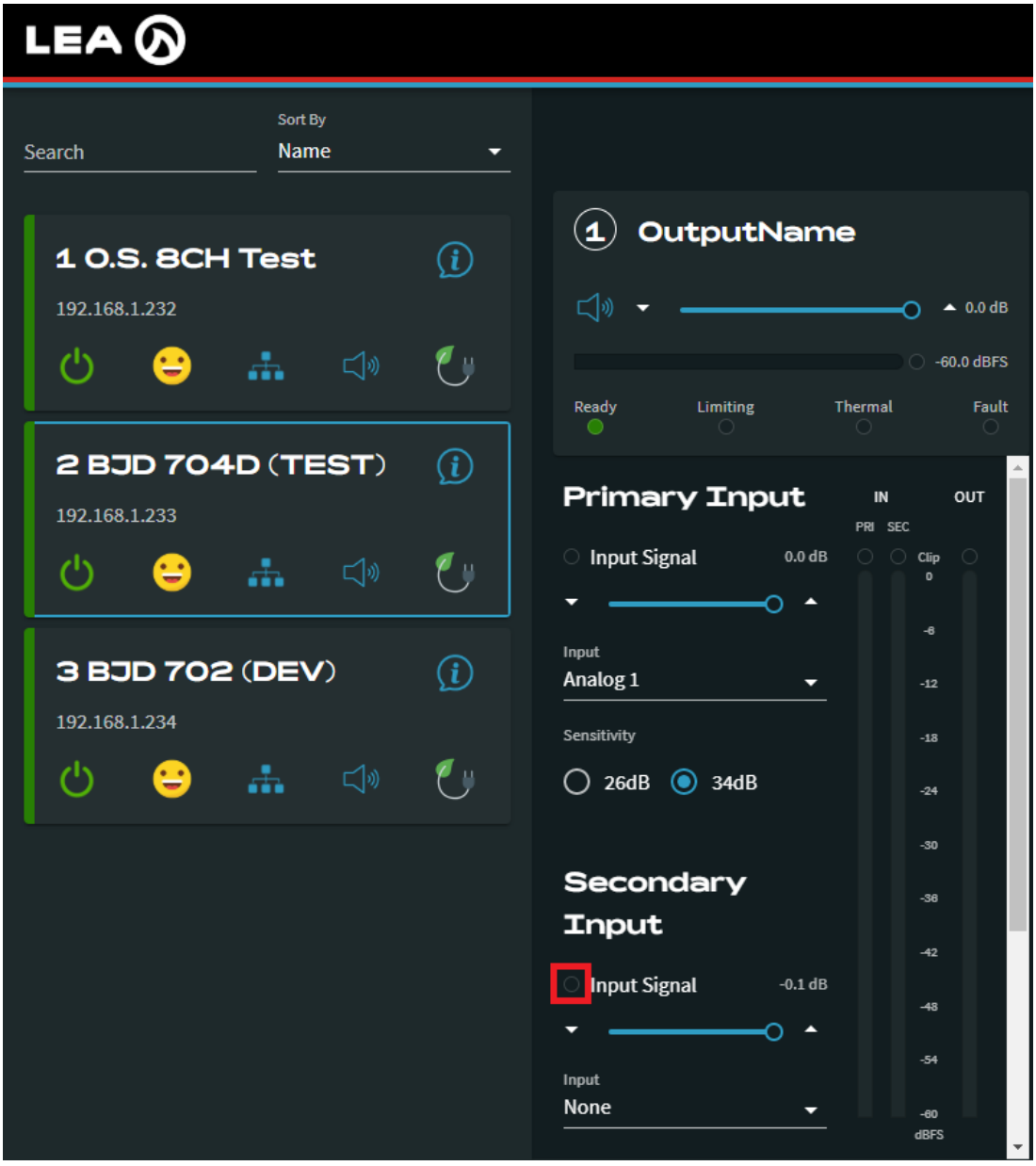
- This will subscribe to the secondary input meter level for channel 1



Secondary Input Signal Present

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/secondarySignalDetect
Values: "true", "false"
Example: get /amp/channels/1/inputSelector/secondarySignalDetect\n

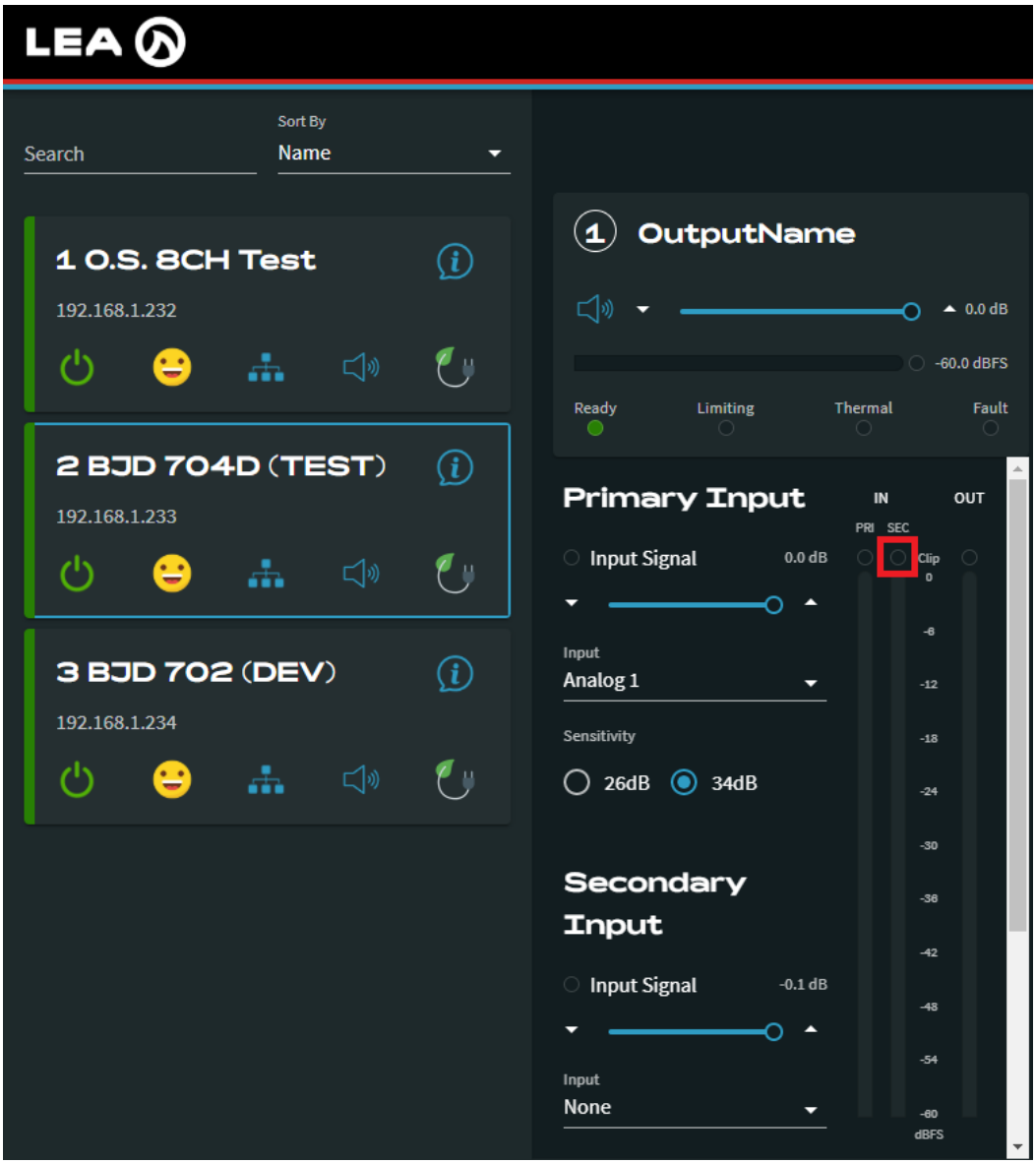
- Response: /amp/channels/1/inputSelector/secondarySignalDetect true\n
- This command asked to get the secondary input signal presence state on Channel 1 and the response back was "true" meaning that signal is present



Secondary Input Signal Clip Indicator

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/secondaryClip
Values: "true", "false"
Example: subscribe /amp/channels/1/inputSelector/secondaryClip\n

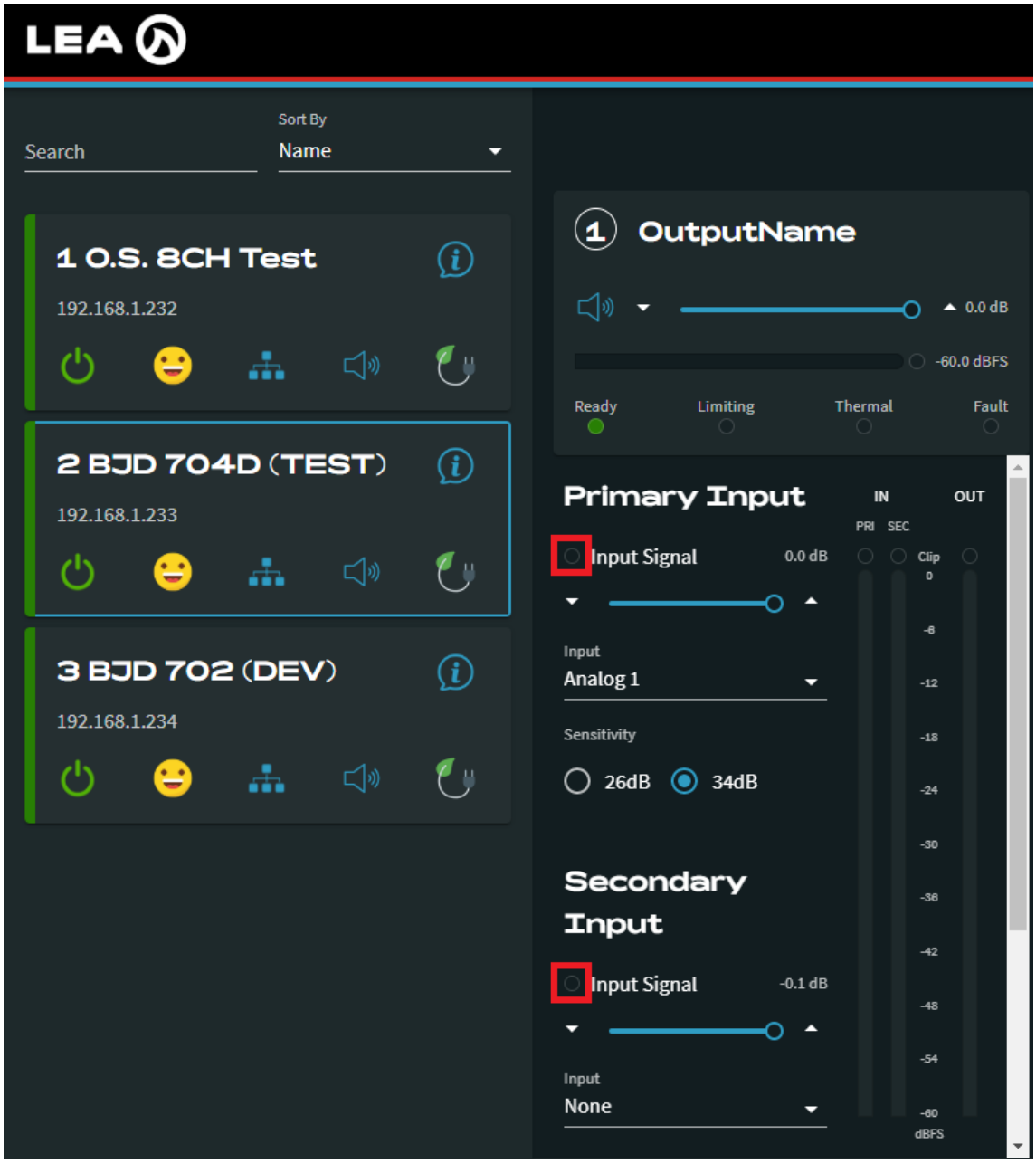
- Response: /amp/channels/1/inputSelector/secondaryClip false\n
- This command asked to subscribe to secondary input signal clip indication state on Channel 1 and the response back was "false" meaning that signal is not clipped, but as this is a subscription any changes here will continue to be reported



Input Signal Present

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/signalDetect
Values: "true", "false"
Example: get /amp/channels/1/inputSelector/signalDetect\n

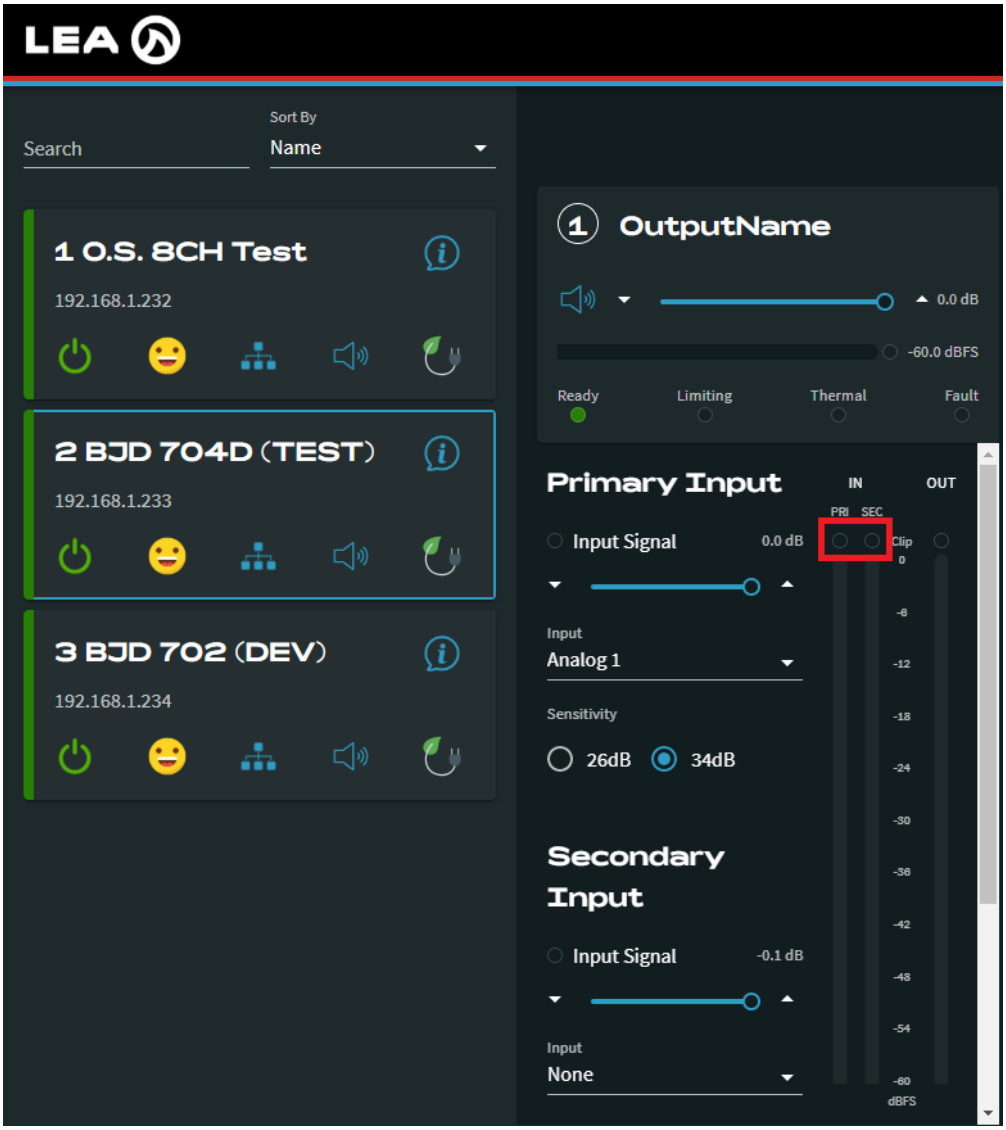
- Response: /amp/channels/1/inputSelector/signalDetect true\n
- This command asked to get the input signal presence state on Channel 1 and the response back was "true" meaning that signal is present (this is an aggregate of both primary and secondary inputs)



Input Signal Clip Indicator

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/clip
Values: "true", "false"
Example: subscribe /amp/channels/1/inputSelector/clip\n

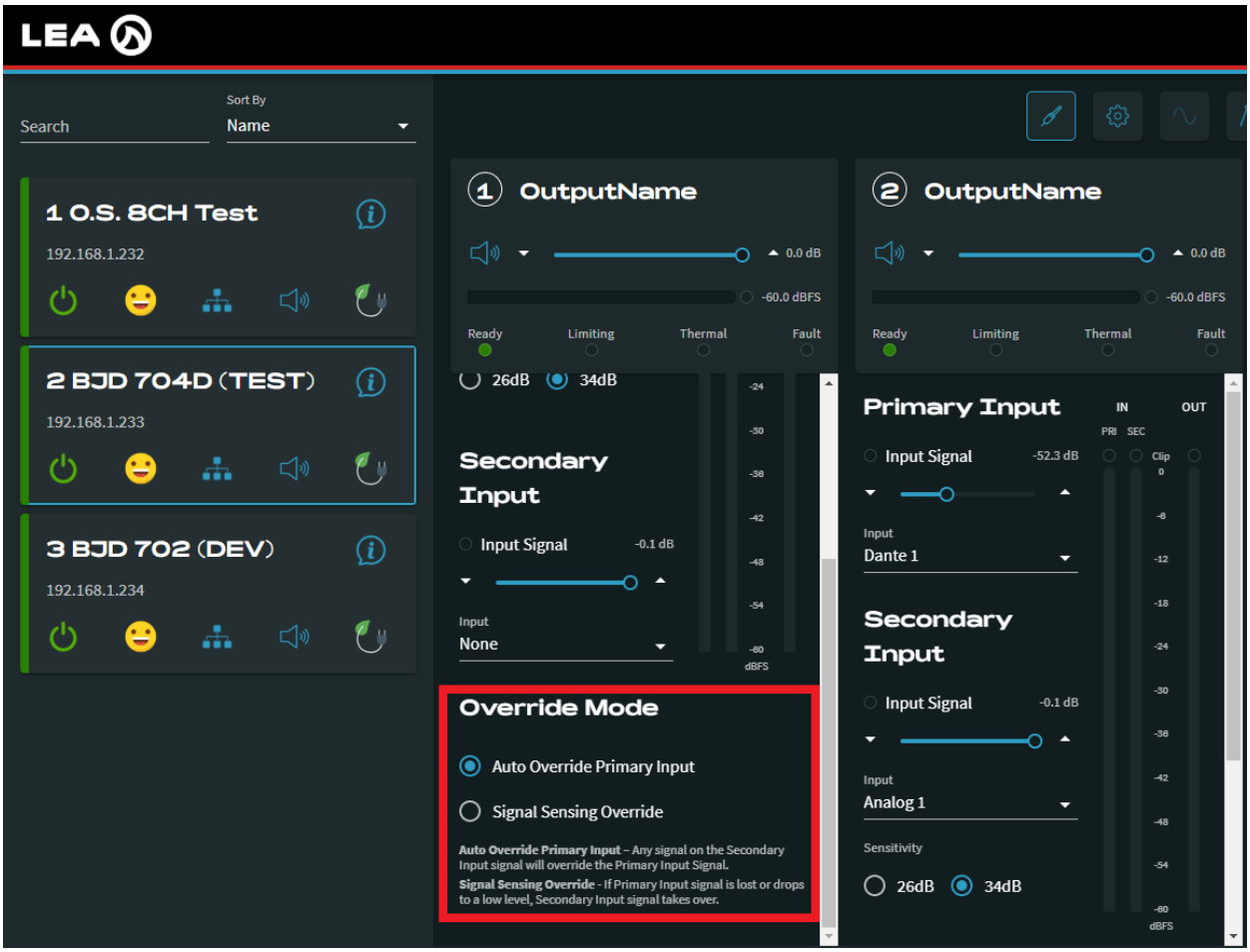
- Response: /amp/channels/1/inputSelector/clip false\n
- This command asked to subscribe to input signal clip indication on Channel 1 and the response back was "false" meaning that signal is not clipped, but as this is a subscription and any changes here will continue to be reported (this is an aggregate of both primary and secondary inputs)



Input Signal Priority Override Mode

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/mode
Values: "Override", "Backup"
Example: set /amp/channels/1/inputSelector/mode "Override"\n

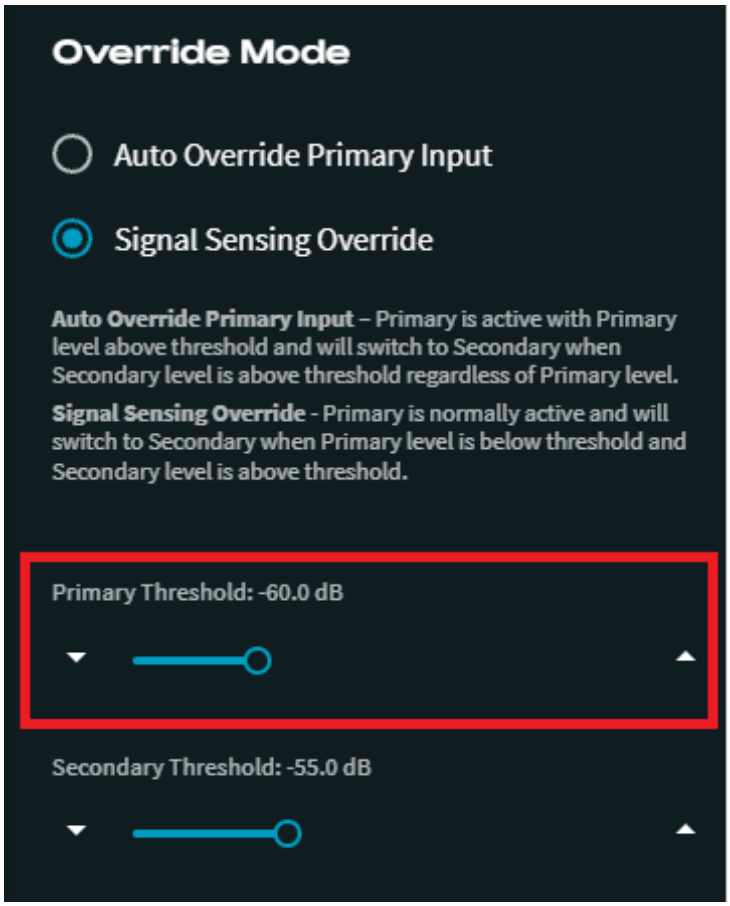
- The command set the priority override mode on channel 1 to "Override"
- "Override" is Auto Override Primary Input
- "Backup" is Signal Sensing Override



Primary Input Signal Override Threshold

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/primaryThreshold
Values: -80 through 0
Example: set /amp/channels/1/inputSelector/primaryThreshold -60\n

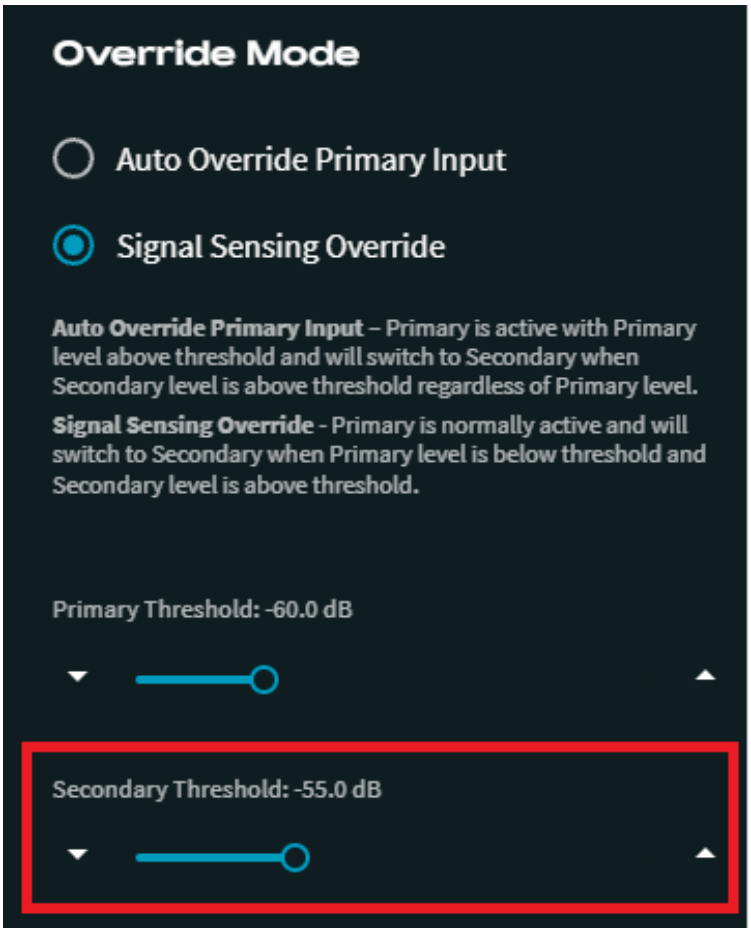
- This command set the secondary override threshold to -60dB



Secondary Input Signal Override Threshold

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/secondaryThreshold
Values: -80 through 0
Example: set /amp/channels/1/inputSelector/secondaryThreshold -55\n

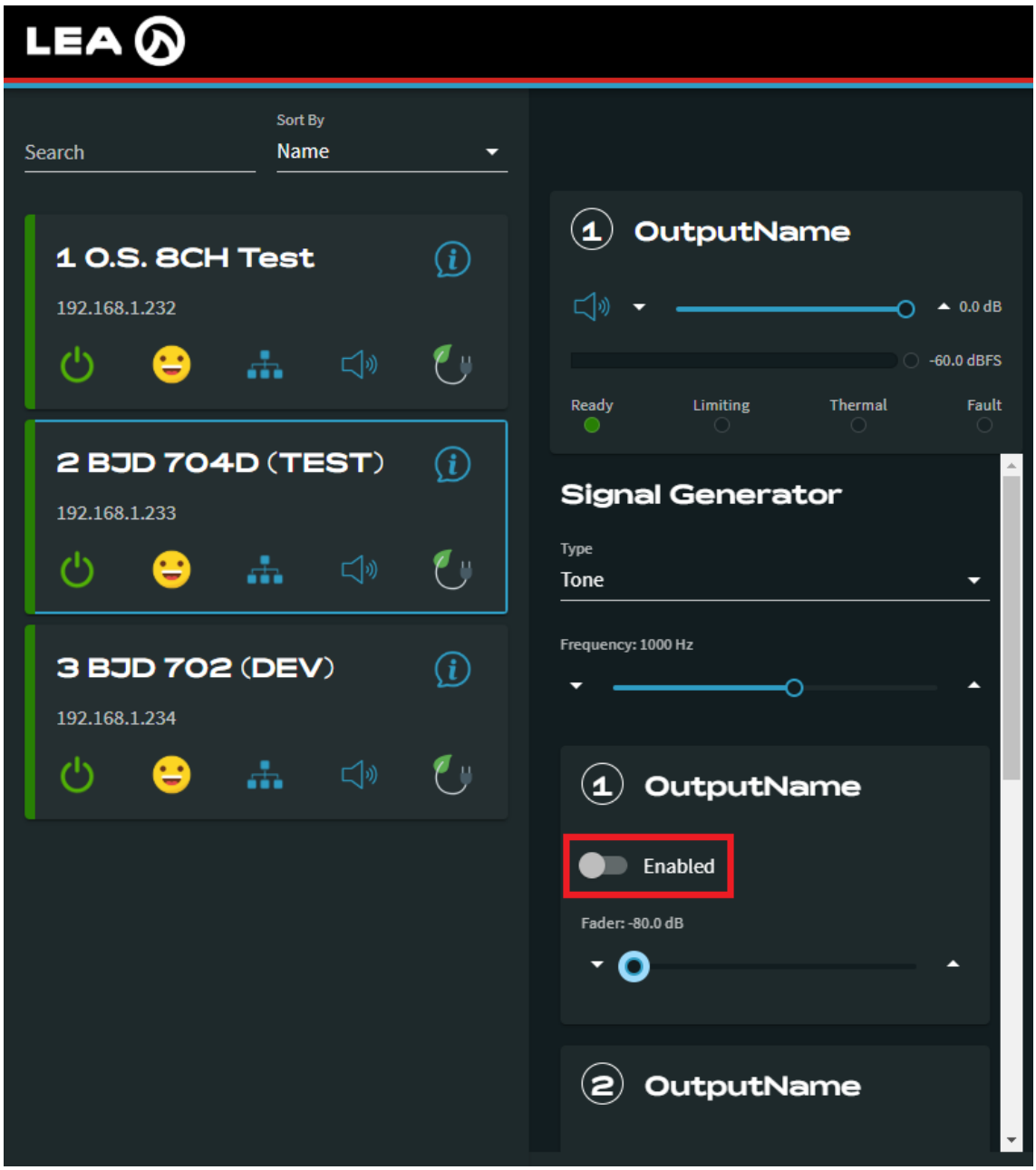
- This command set the secondary override threshold to -55dB



Signal Generator Channel Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/signalGeneratorEnable
Values: "true", "false"
Example: set /amp/channels/1/inputSelector/signalGeneratorEnable "true"\n

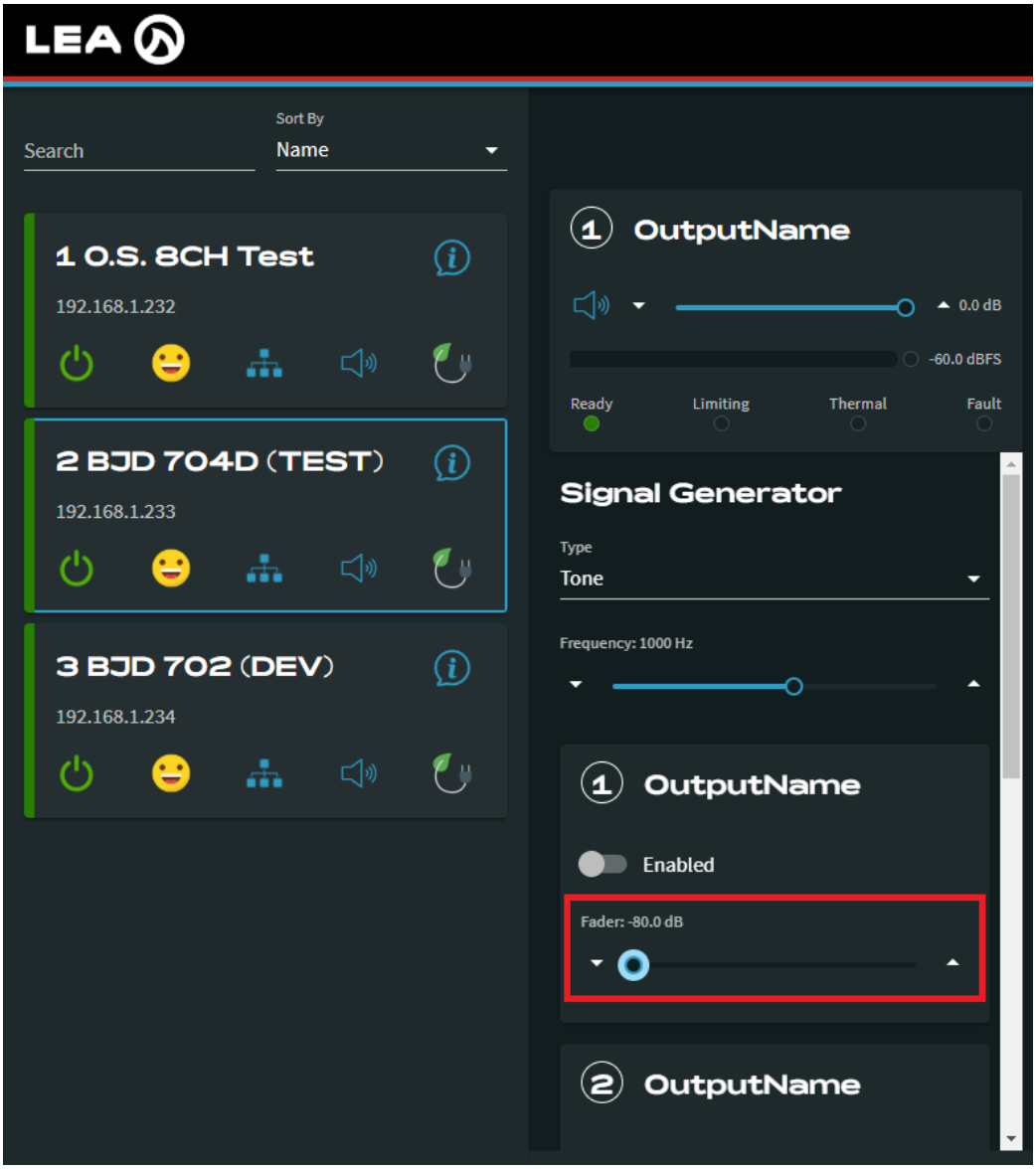
- This command enabled the signal generator on Channel 1



Signal Generator Channel Fader

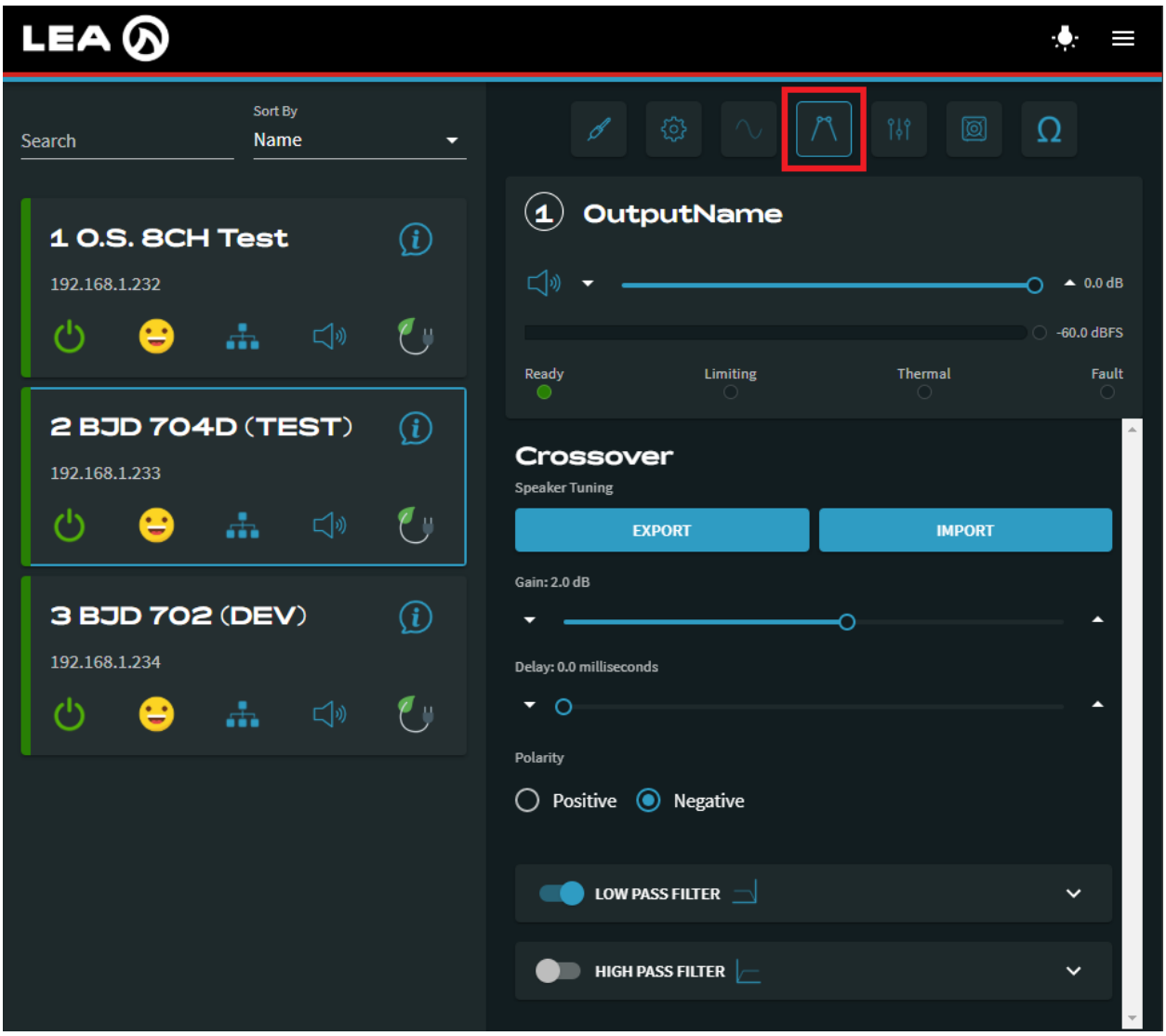
Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/signalGeneratorFader
Values: -80 through 0 dB
Example: set /amp/channels/1/inputSelector/signalGeneratorFader -20.0\n

- This command set the signal generator fader to -20 dB on Channel 1



Amplifier Channels Crossover

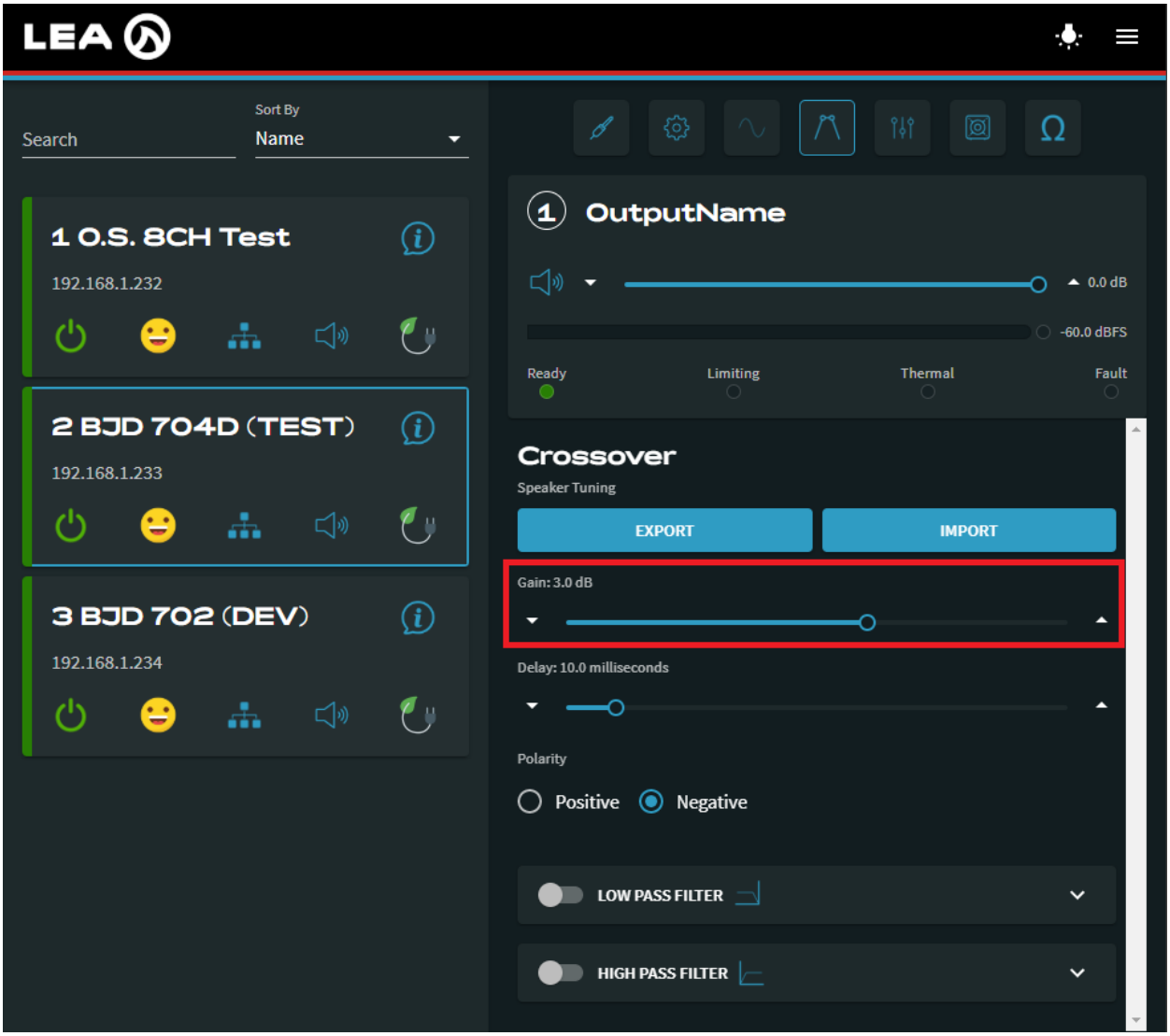
Click on this button to navigate to the Crossover section of the DSP relevant to this API section



Crossover Gain

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/crossover/bandGainAndDelay/gain
Values: -15.0 through 15.0
Example: set /amp/channels/1/crossover/bandGainAndDelay/gain 3.0\n

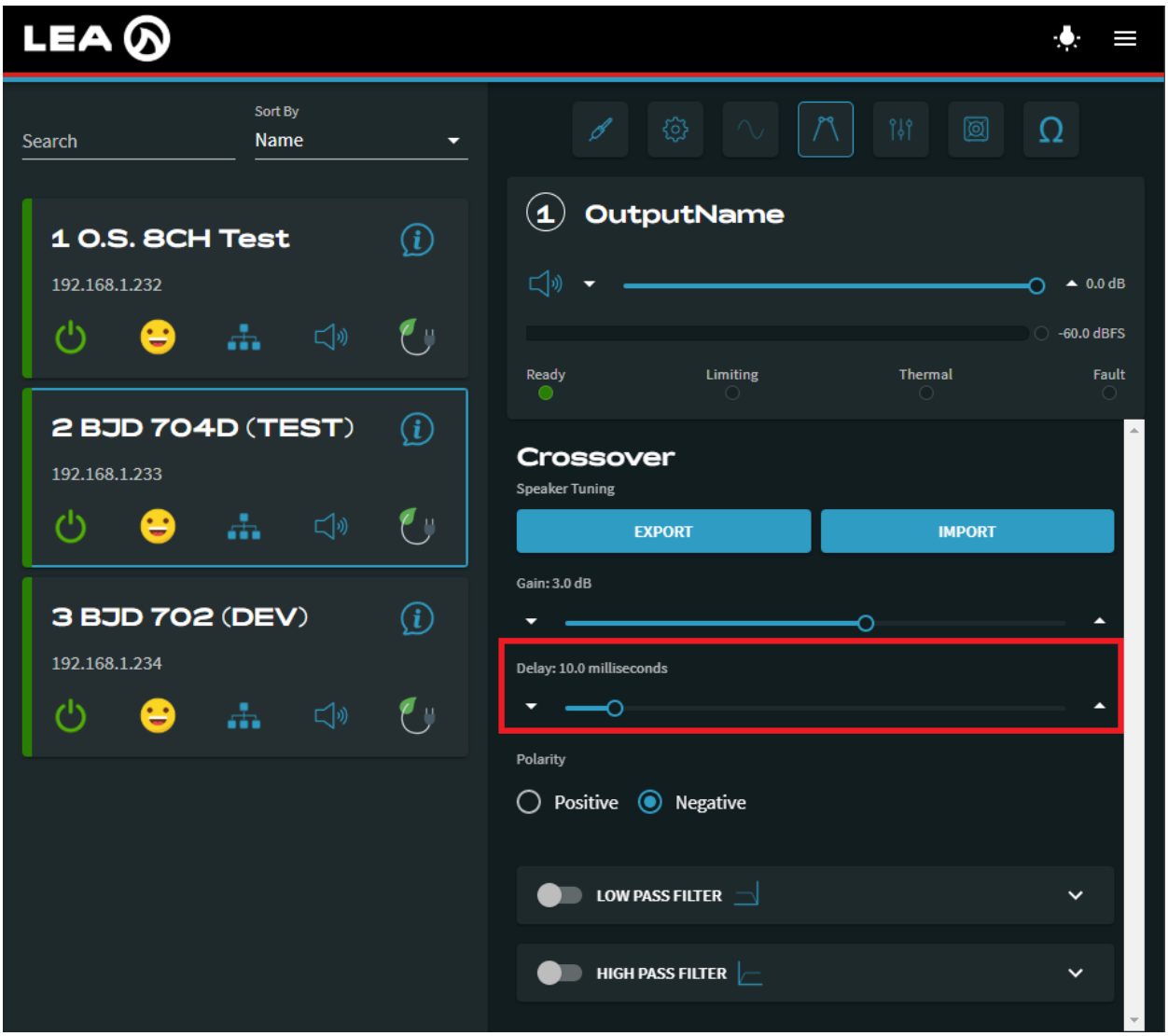
- This command set the crossover gain on channel 1 to 3.0 dB



Crossover Delay

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/crossover/bandGainAndDelay/delay
Values: 0.0 through 100.0
Example: set /amp/channels/1/crossover/bandGainAndDelay/delay 10.0\n

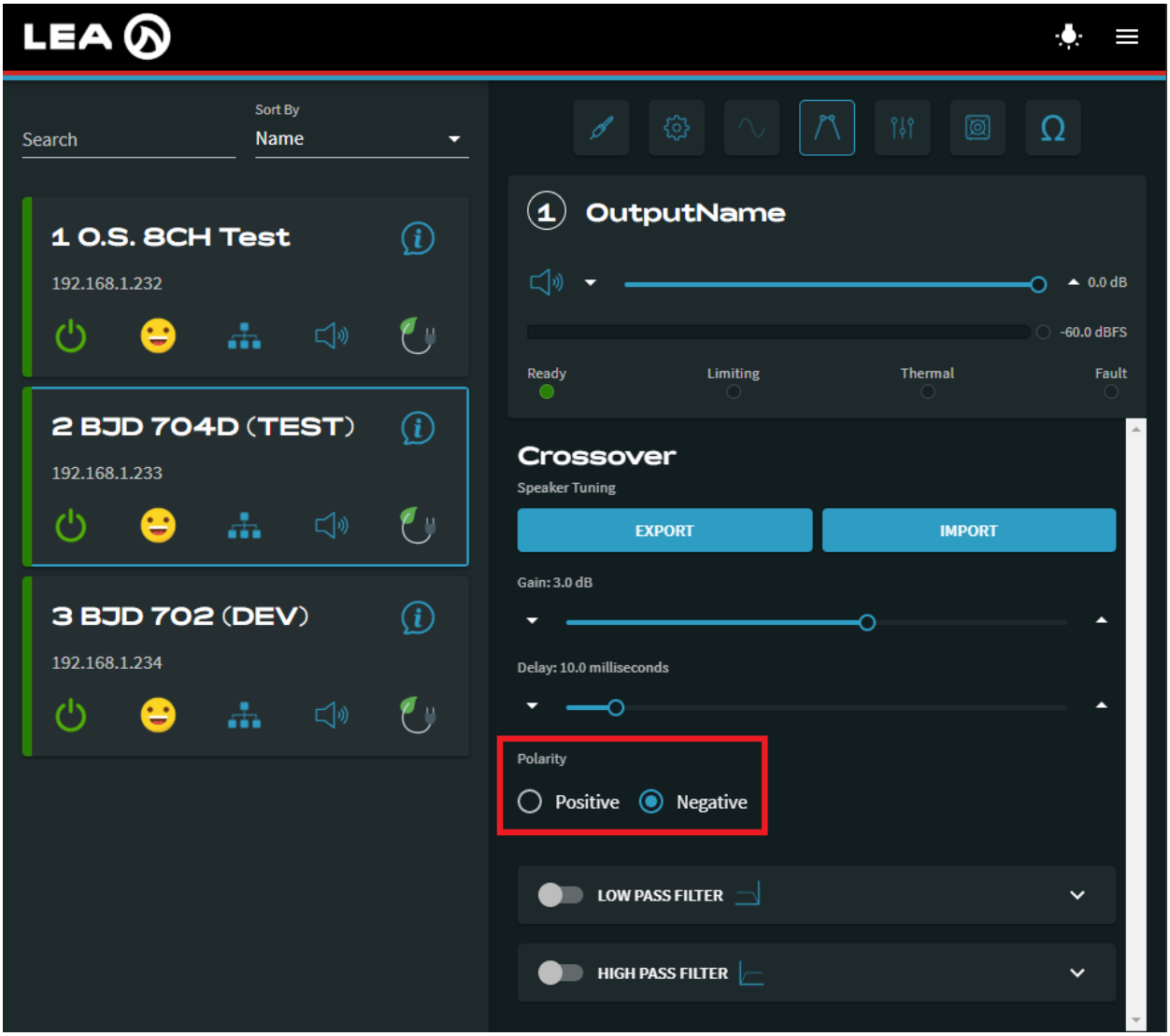
- This command set the crossover delay on channel 1 to 10.0 ms



Crossover Polarity

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/crossover/bandGainAndDelay/invert
Values: "true", "false"
Example: set /amp/channels/1/crossover/bandGainAndDelay/invert "true"\n

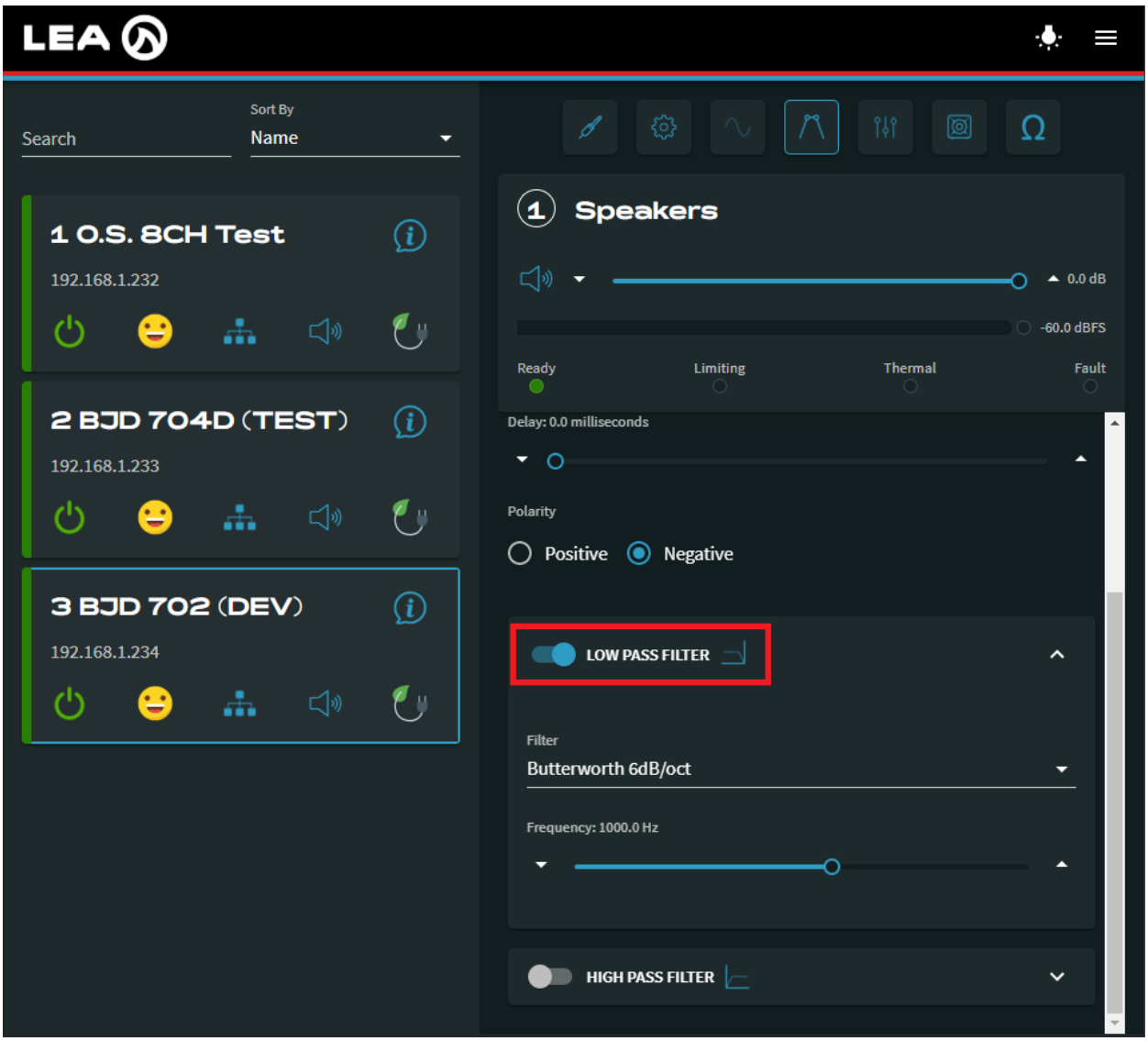
- This command set the crossover polarity to negative on channel 1



Crossover Low Pass Filter Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/crossover/lowPassFilter/enable
Values: "true", "false"
Example: set /amp/channels/1/crossover/lowPassFilter/enable "true"\n

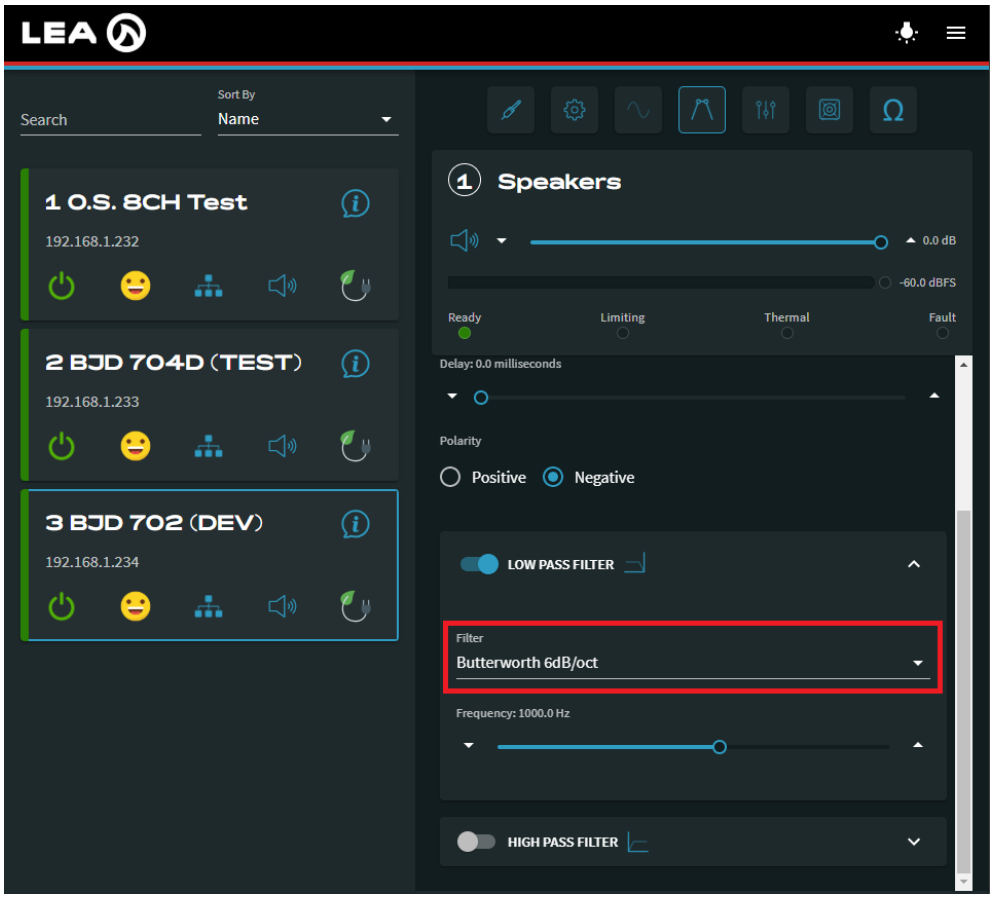
- This command enabled the Low Pass Filter on channel 1



Crossover Low Pass Filter Type

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/crossover/lowPassFilter/type
Values:
"Butterworth 6dB/oct"
"Butterworth 12dB/oct"
"Butterworth 18dB/oct"
"Butterworth 24dB/oct"
"Butterworth 48dB/oct"
"Linkwitz-Riley 24dB/oct"
"Linkwitz-Riley 48dB/oct"
"Bessel 6dB/oct"
"Bessel 12dB/oct"
"Bessel 18dB/oct"
"Bessel 24dB/oct"
"Bessel 48dB/oct"
Example: set /amp/channels/1/crossover/lowPassFilter/type "Butterworth 6dB/oct"\n

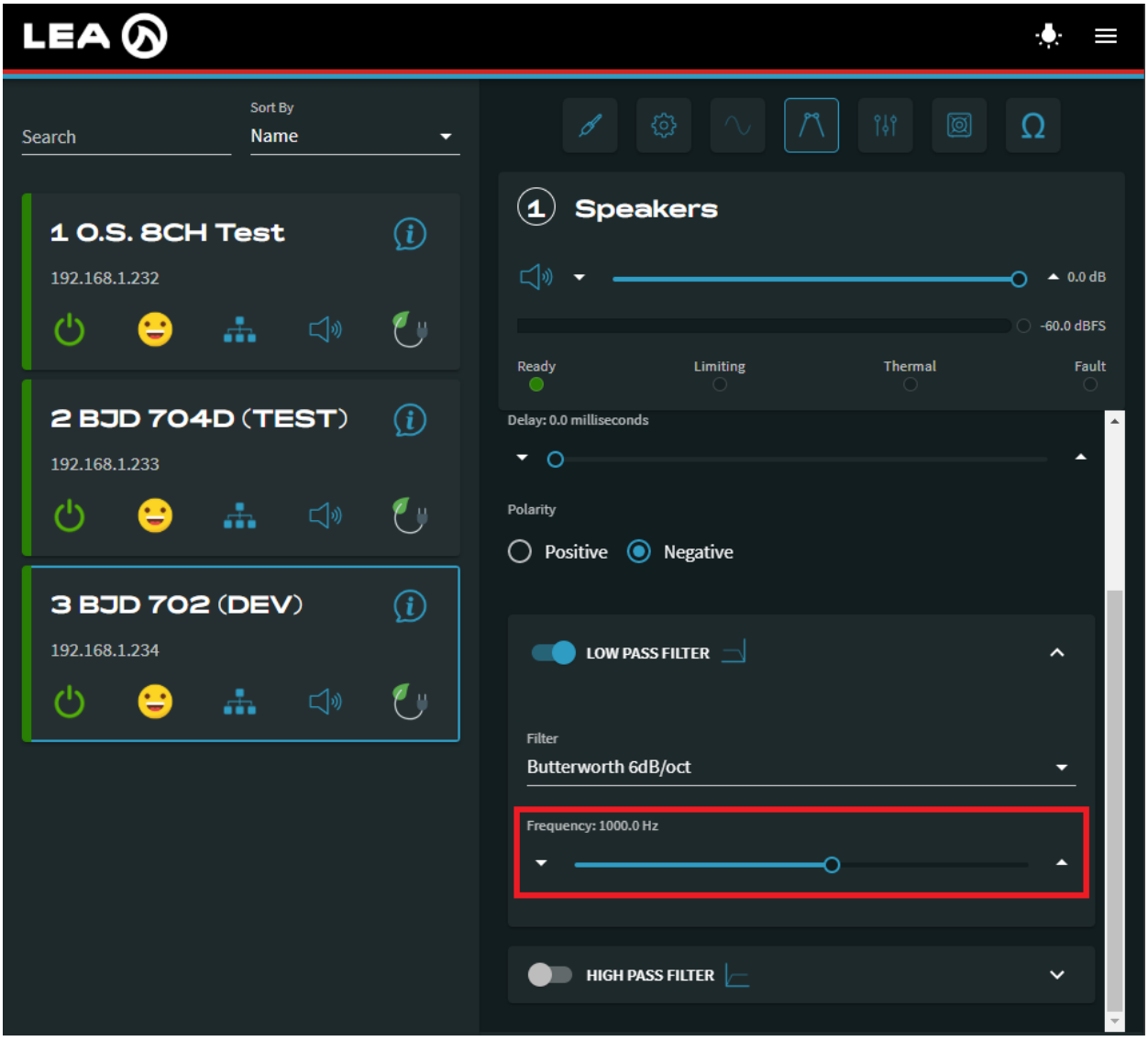
- This command set the Low Pass Filter type to Butterworth 6dB/oct on channel 1



Crossover Low Pass Filter Frequency

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/crossover/lowPassFilter/frequency
Values: 20 through 20000
Example: set /amp/channels/1/crossover/lowPassFilter/frequency 1000\n

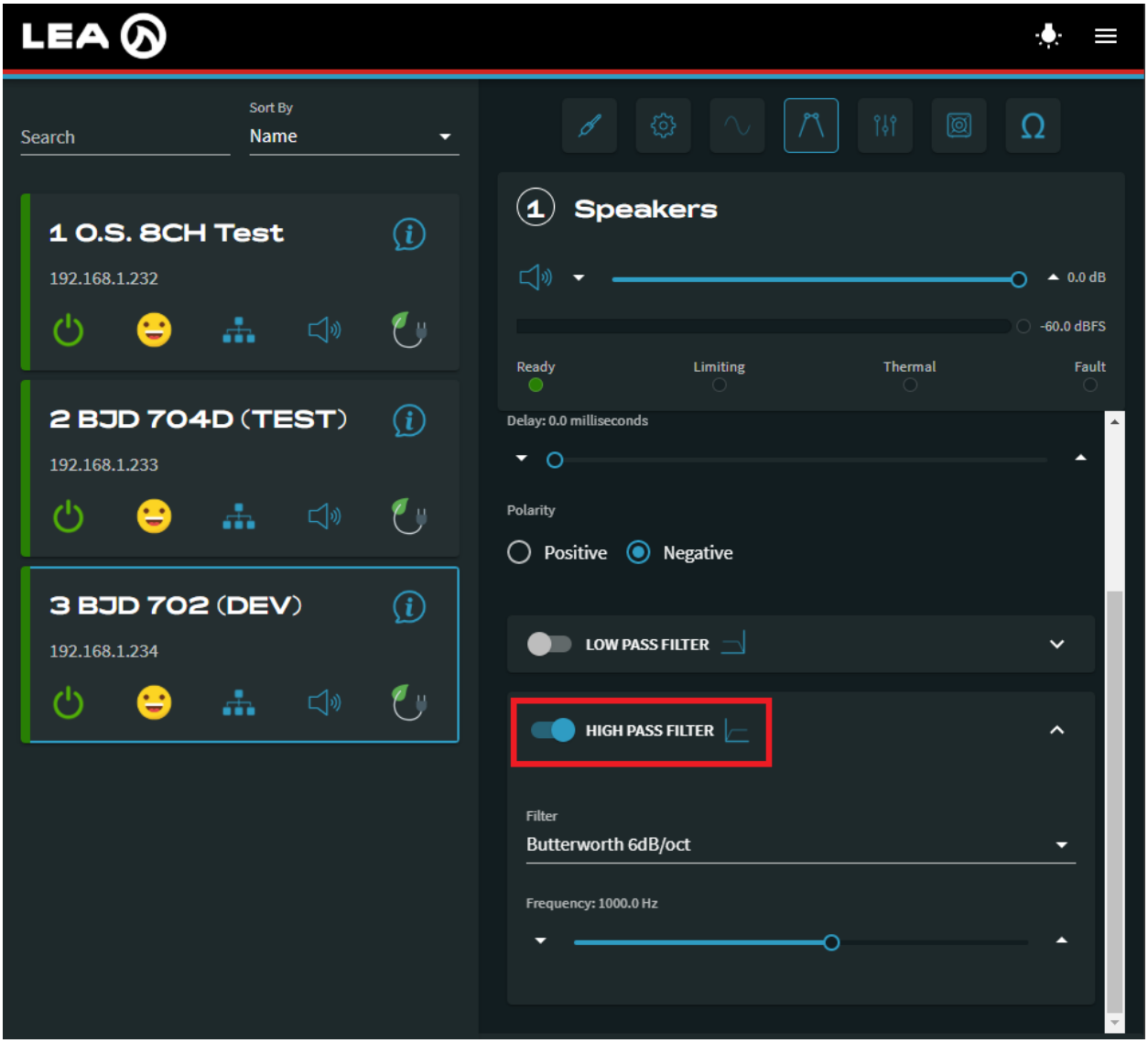
- This command set the Low Pass Filter Frequency on channel 1 to 1000 Hz



Crossover High Pass Filter Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/crossover/highPassFilter/enable
Values: "true", "false"
Example: set /amp/channels/1/crossover/highPassFilter/enable "true"\n

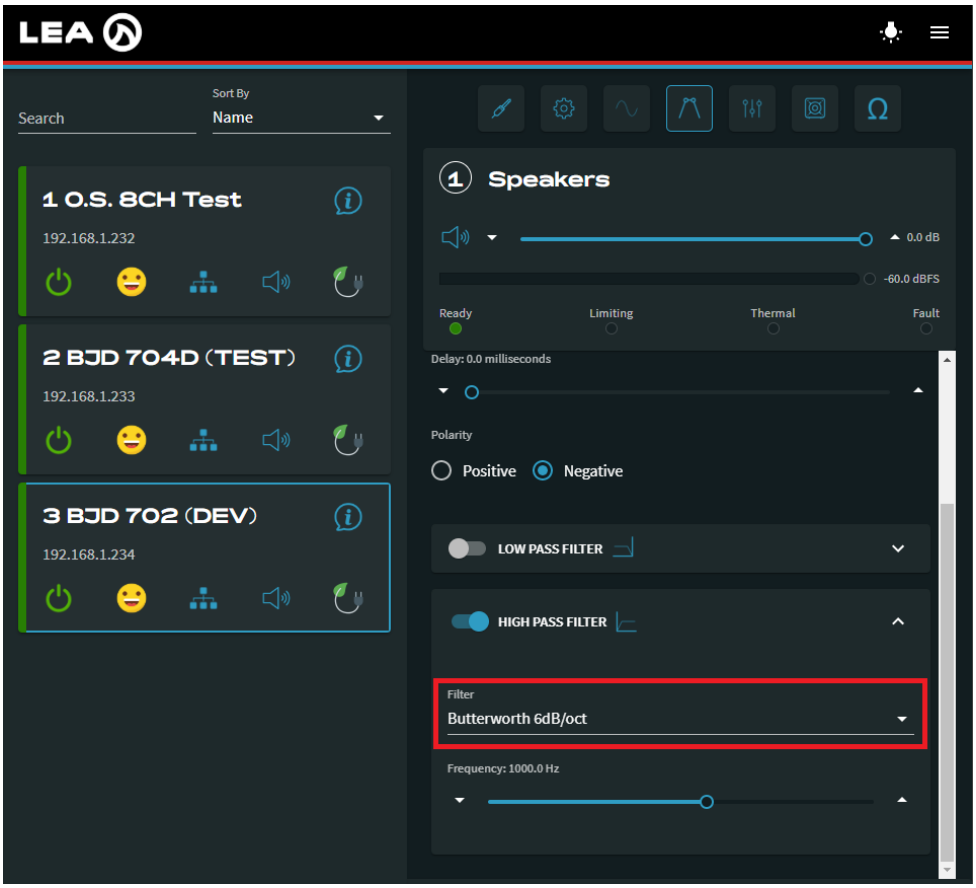
- This command enabled the High Pass Filter on channel 1



Crossover High Pass Filter Type

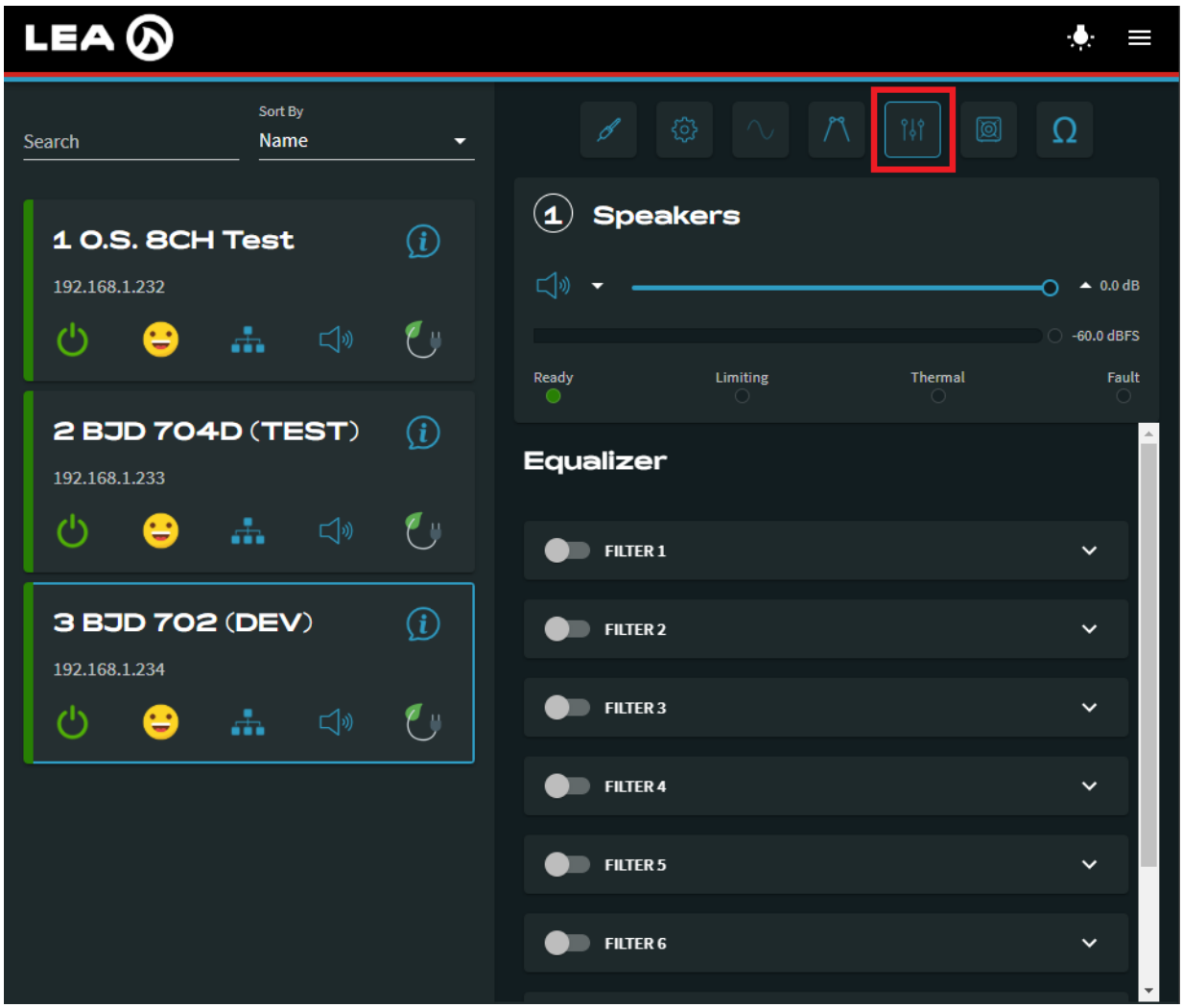
Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/crossover/highPassFilter/type
Values:
"Butterworth 6dB/oct"
"Butterworth 12dB/oct"
"Butterworth 18dB/oct"
"Butterworth 24dB/oct"
"Butterworth 48dB/oct"
"Linkwitz-Riley 24dB/oct"
"Linkwitz-Riley 48dB/oct"
"Bessel 6dB/oct"
"Bessel 12dB/oct"
"Bessel 18dB/oct"
"Bessel 24dB/oct"
"Bessel 48dB/oct"
Example: set /amp/channels/1/crossover/lowPassFilter/type "Butterworth 6dB/oct"\n

- This command set the Low Pass Filter type to Butterworth 6dB/oct on channel 1



Amplifier Channels Output EQ

Click on this button to navigate to the EQ section of the DSP relevant to this API section



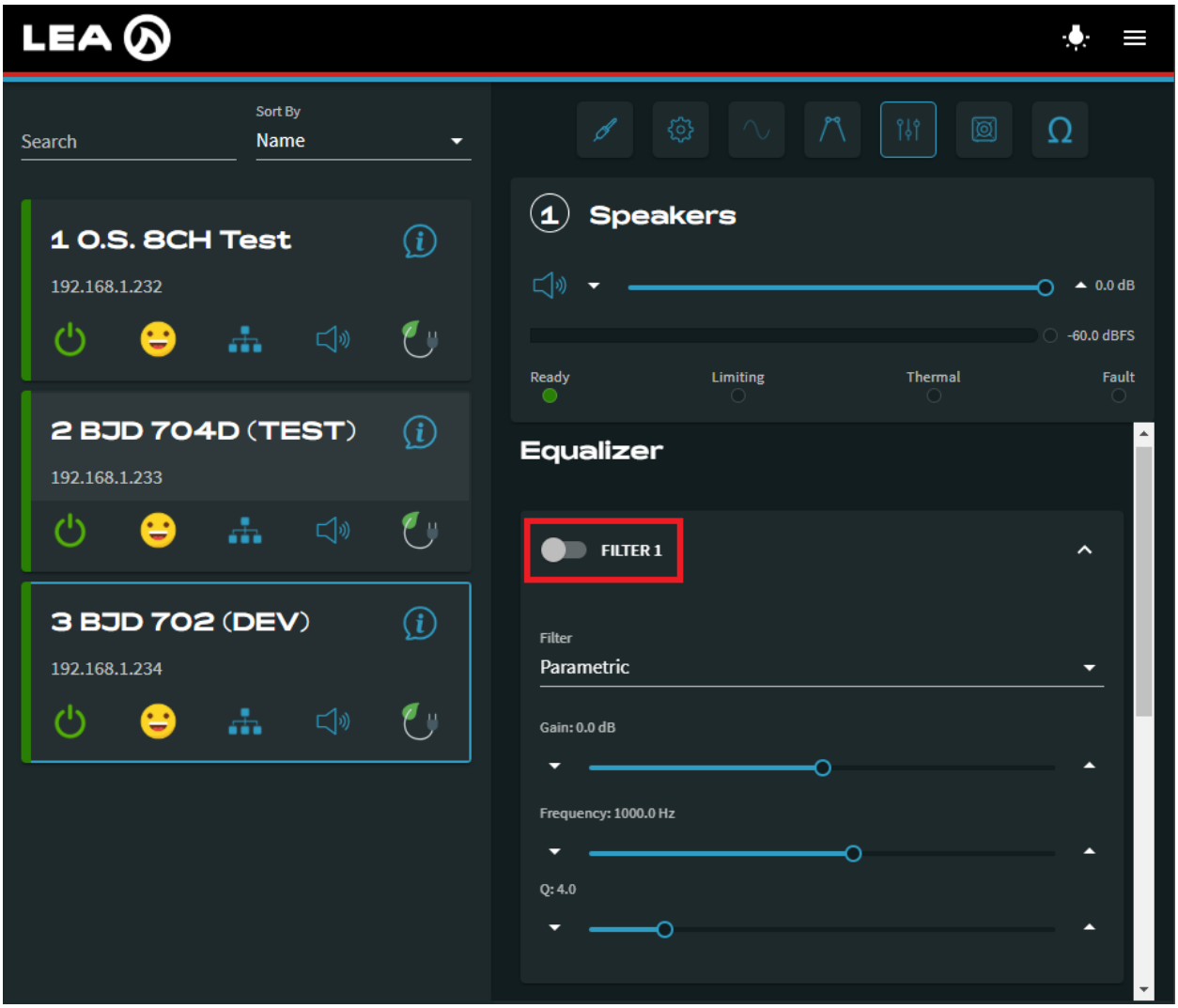
Output EQ Filter Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/outputEqFilters/*/enable

- x is the desired channel number and * is the desired filter number

Values: "true", "false"
Example: set /amp/channels/1/outputEqFilters/1/enable "true"\n

- This command enabled the EQ Filter 1 on Channel 1



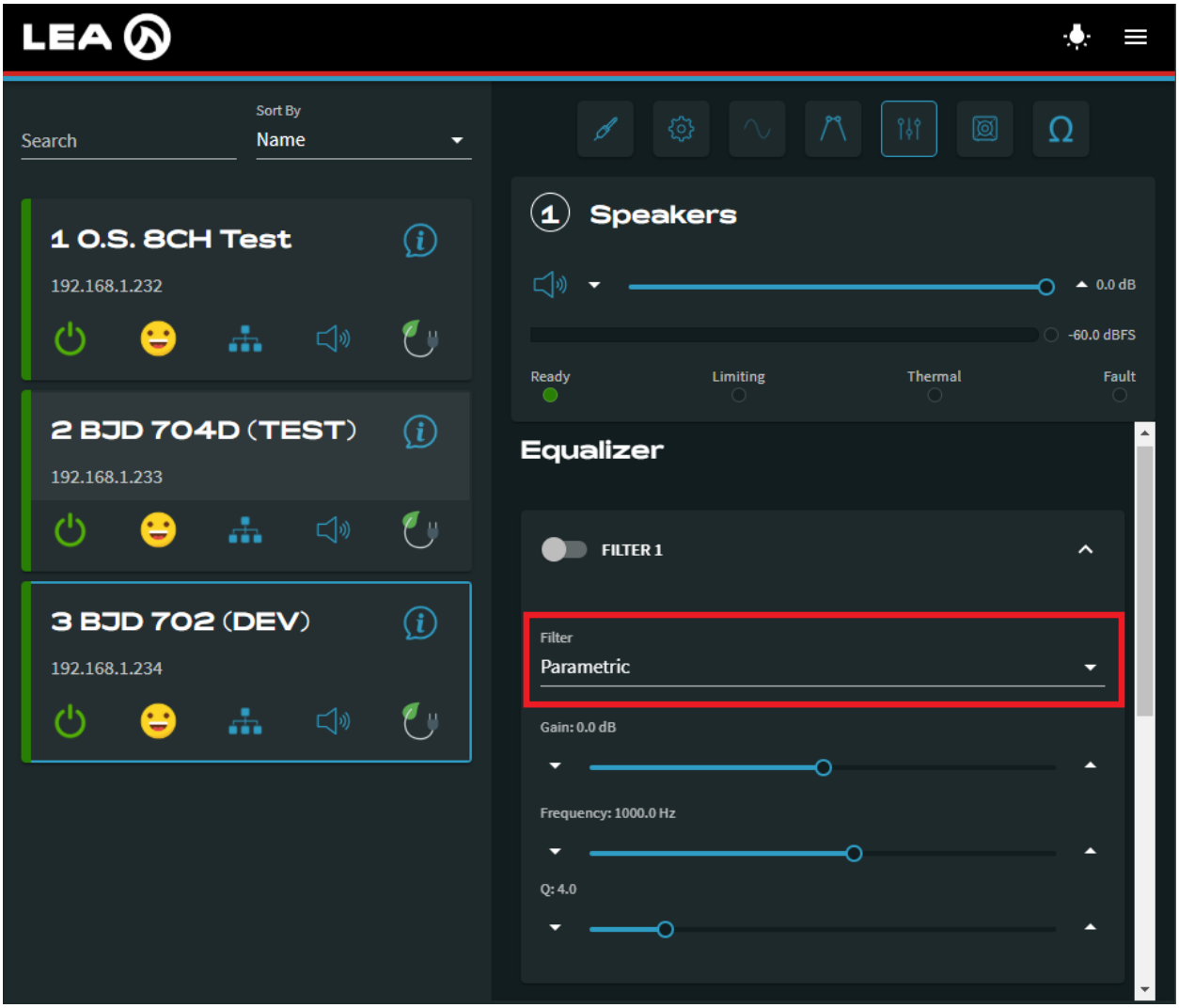
Output EQ Filter Type

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/outputEqFilters/*/type

- x is the desired channel number and * is the desired filter number

Values:
"Parametric"
"LP Shelf 6dB/oct"
"HP Shelf 6dB/oct"
"LP Shelf 12dB/oct"
"HP Shelf 12dB/oct"
Example: set /amp/channels/1/outputEqFilters/1/type "Parametric"\n

- This command set the filter type on EQ Filter 1 on Channel 1 to Parametric



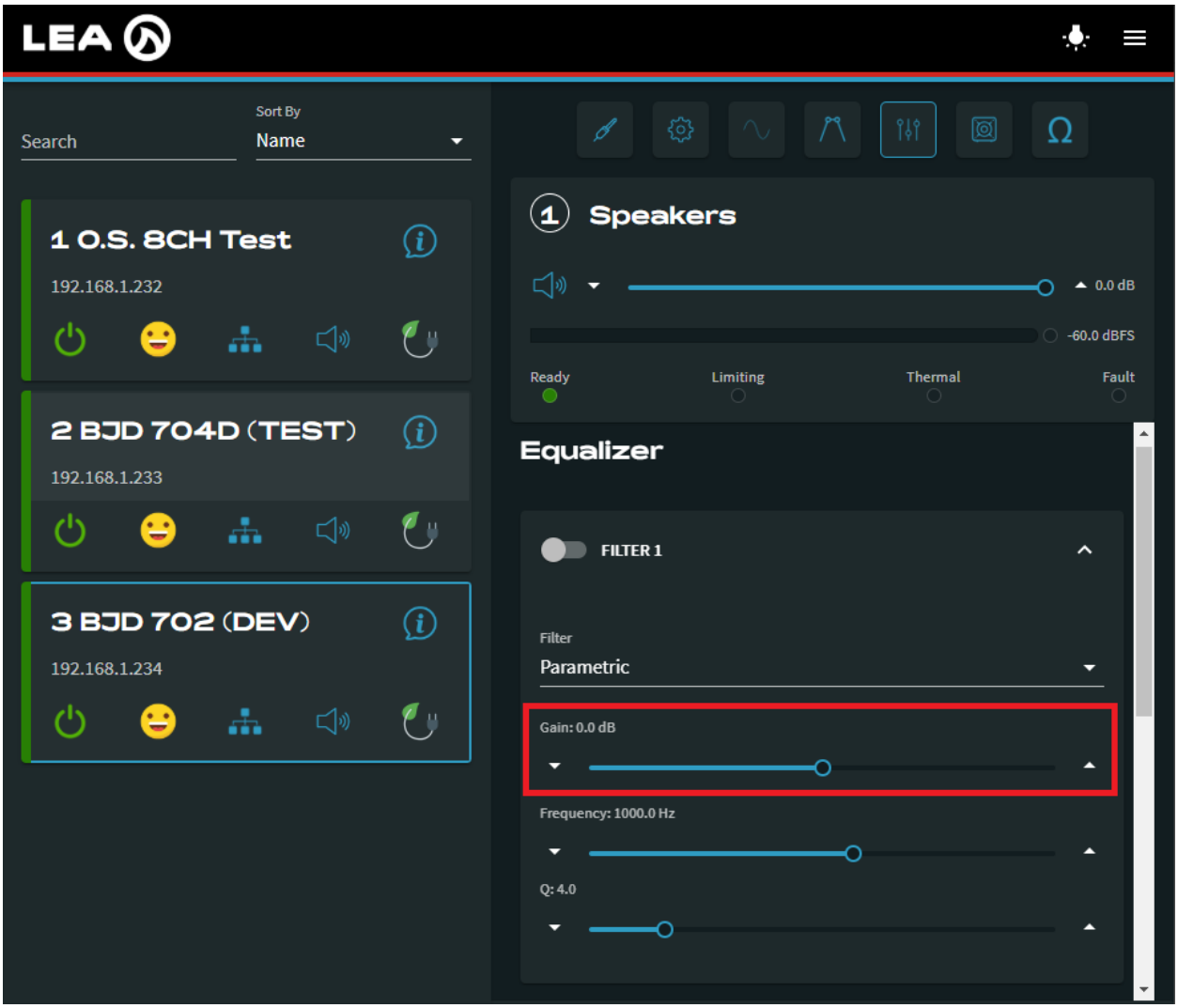
Output EQ Filter Gain

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/outputEqFilters/*/gain

- x is the desired channel number and * is the desired filter number

Values: -15.0 through 15.0
Example: set /amp/channels/1/outputEqFilters/1/gain 3.0\n

- This command set the gain on EQ Filter 1 on Channel 1 to 3.0 dB



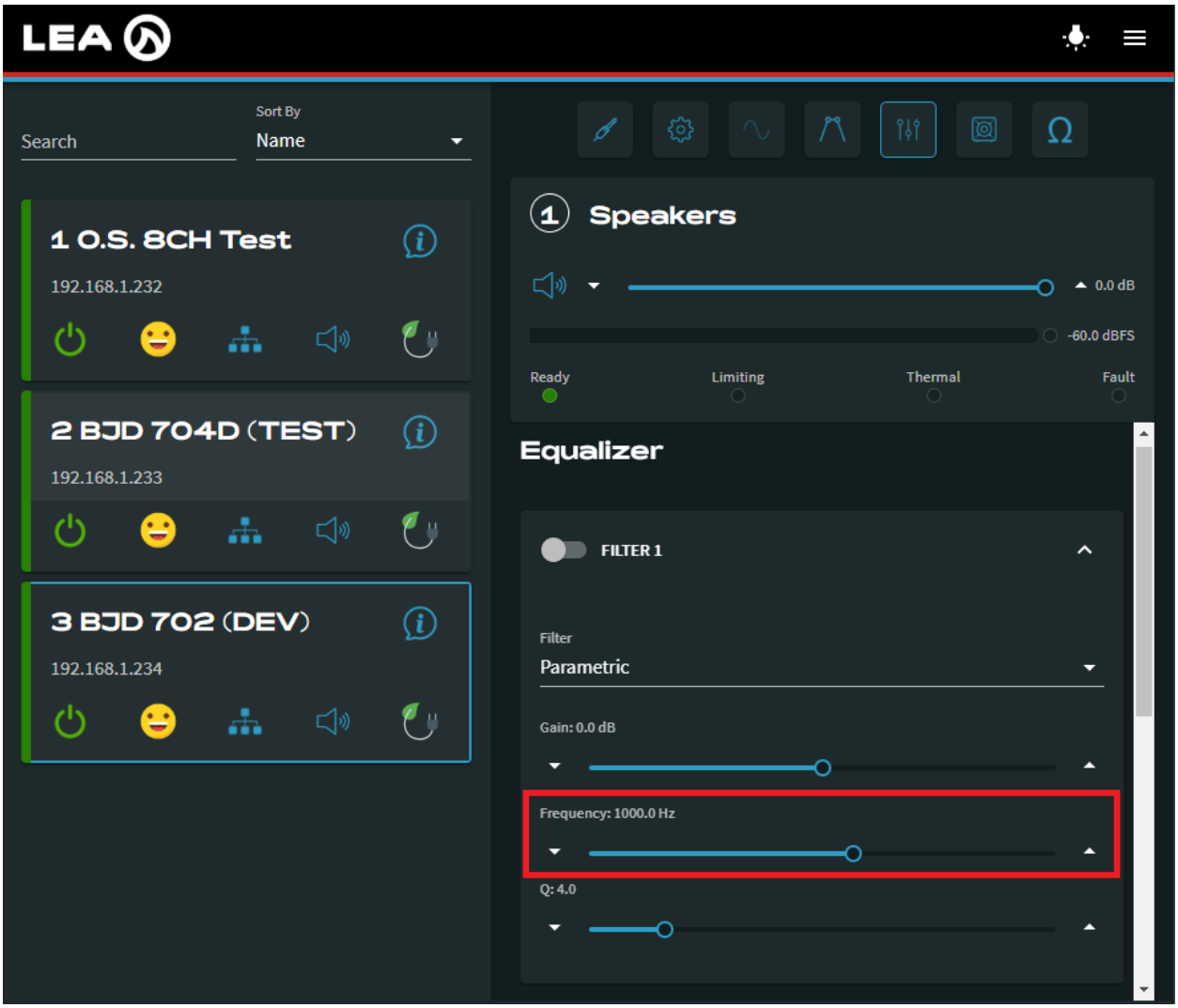
Output EQ Filter Frequency

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/outputEqFilters/*/frequency

- x is the desired channel number and * is the desired filter number

Values: 20 through 20000
Example: set /amp/channels/1/outputEqFilters/1/frequency 1000\n

- This command set the frequency on EQ Filter 1 on Channel 1 to 1000 Hz



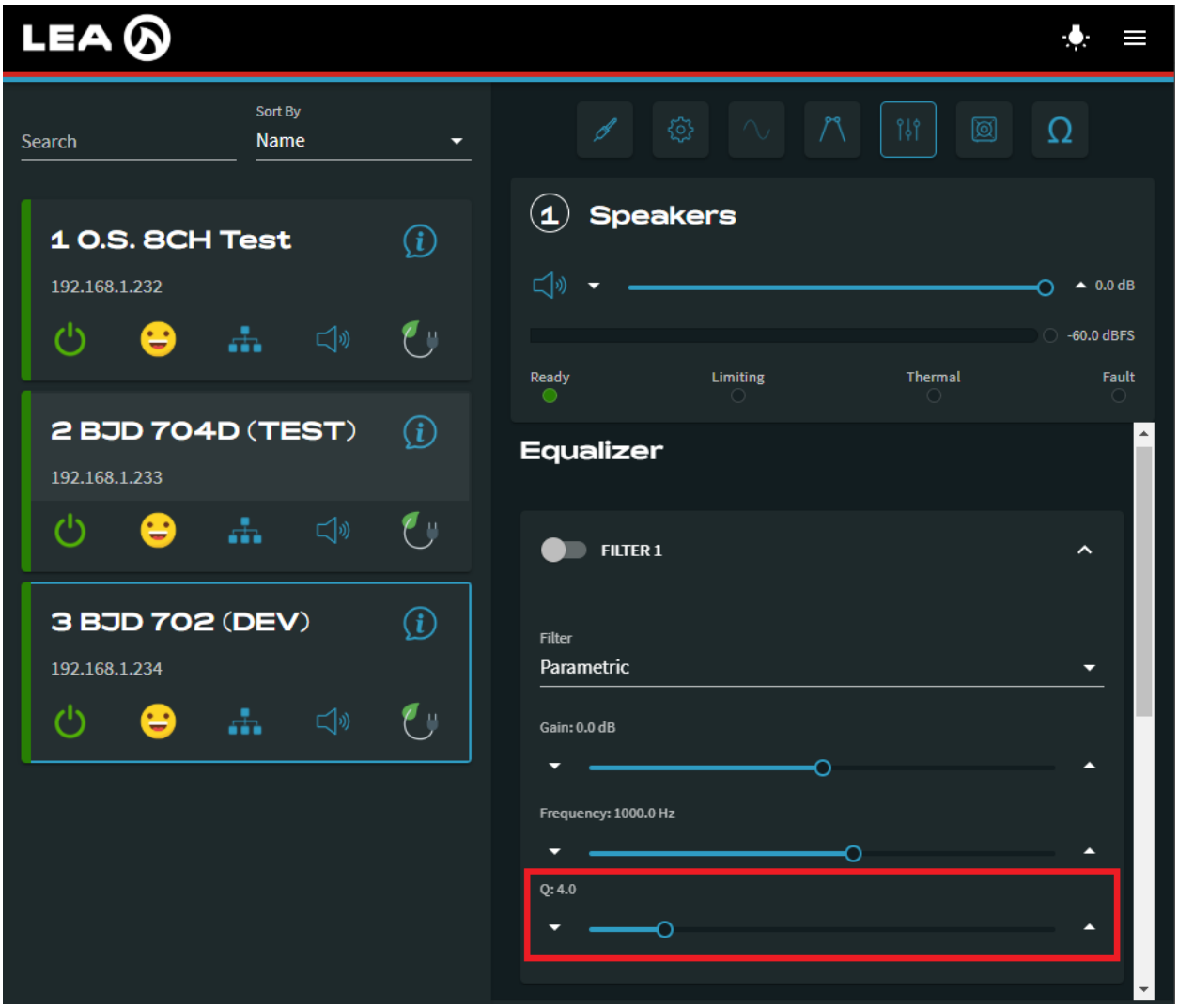
Output EQ Filter Q

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/outputEqFilters/*/q

- x is the desired channel number and * is the desired filter number

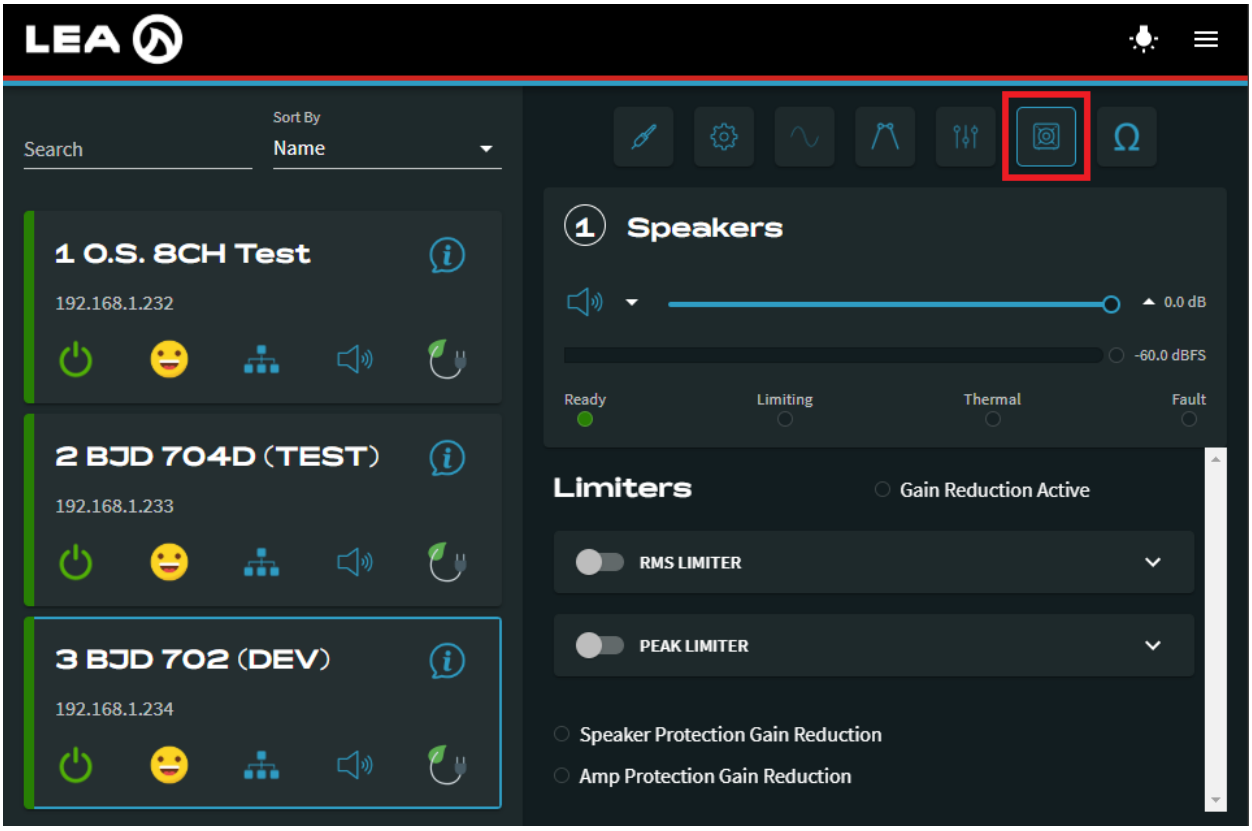
Values: 0.1 through 24.0
Example: set /amp/channels/1/outputEqFilters/1/q 4.0\n

- This command set the Q on EQ Filter 1 on Channel 1 to 4.0



Amplifier Channels RMS Limiter and Peak Limiter

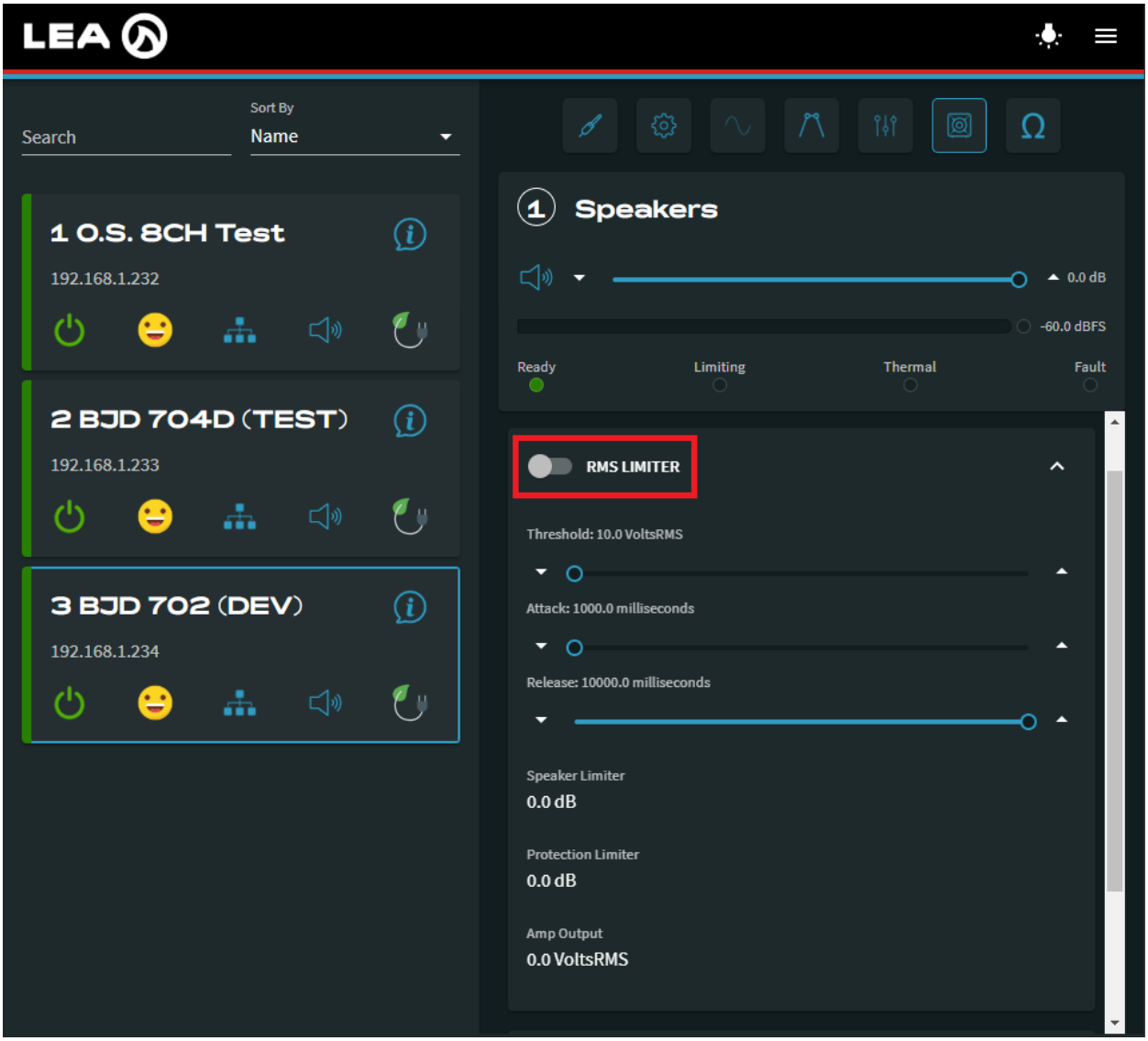
Click on this button to navigate to the Limiter section of the DSP relevant to this API section



RMS Limiter Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/rmsLimiter/enable
Values: "true", "false"
Example: set /amp/channels/1/rmsLimiter/enable "false"\n

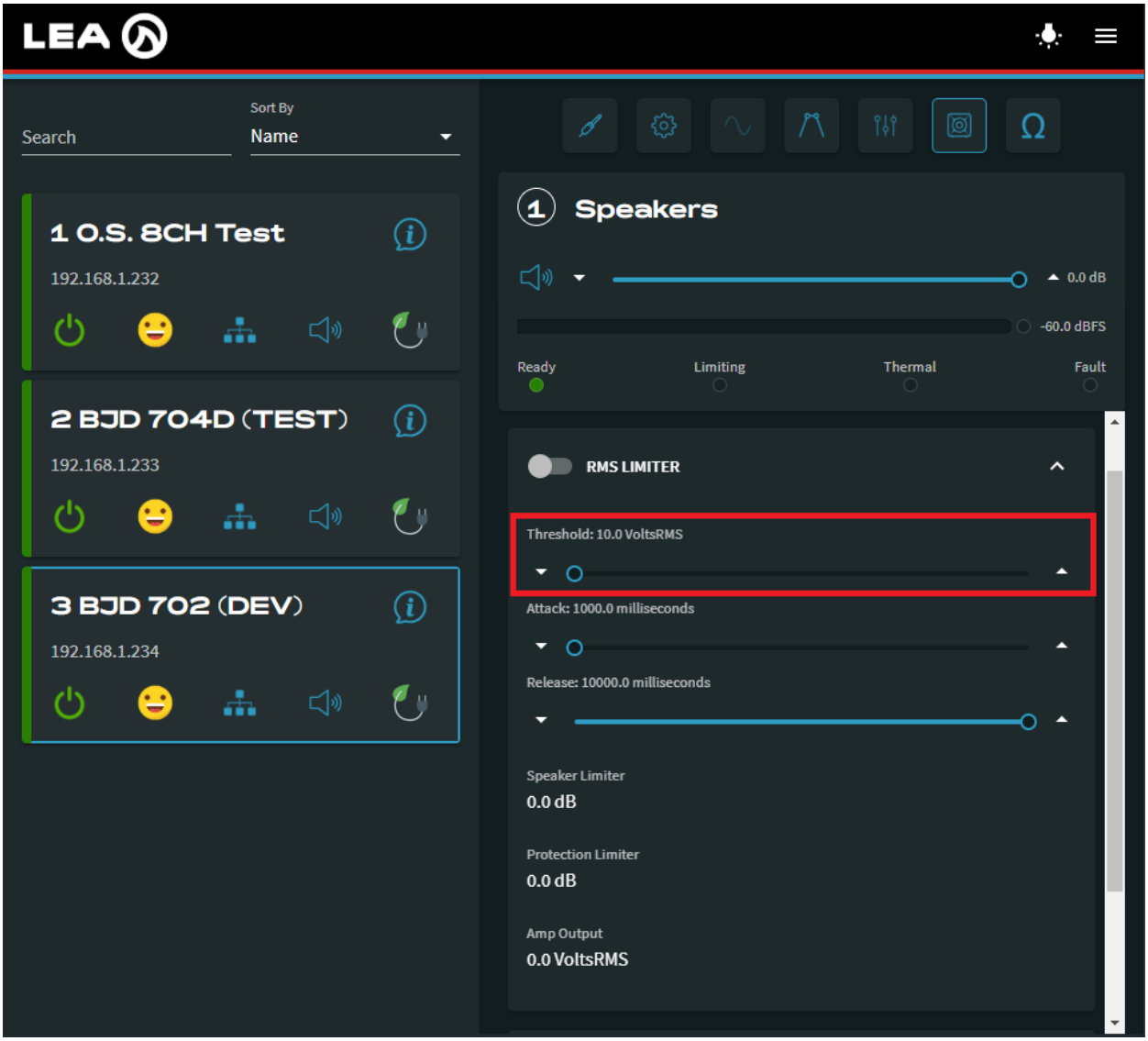
- This command disabled the RMS Limiter on channel 1



RMS Limiter Threshold

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/rmsLimiter/threshold
Values: 10.0 through 140.0
Example: set /amp/channels/1/rmsLimiter/threshold 70.0\n

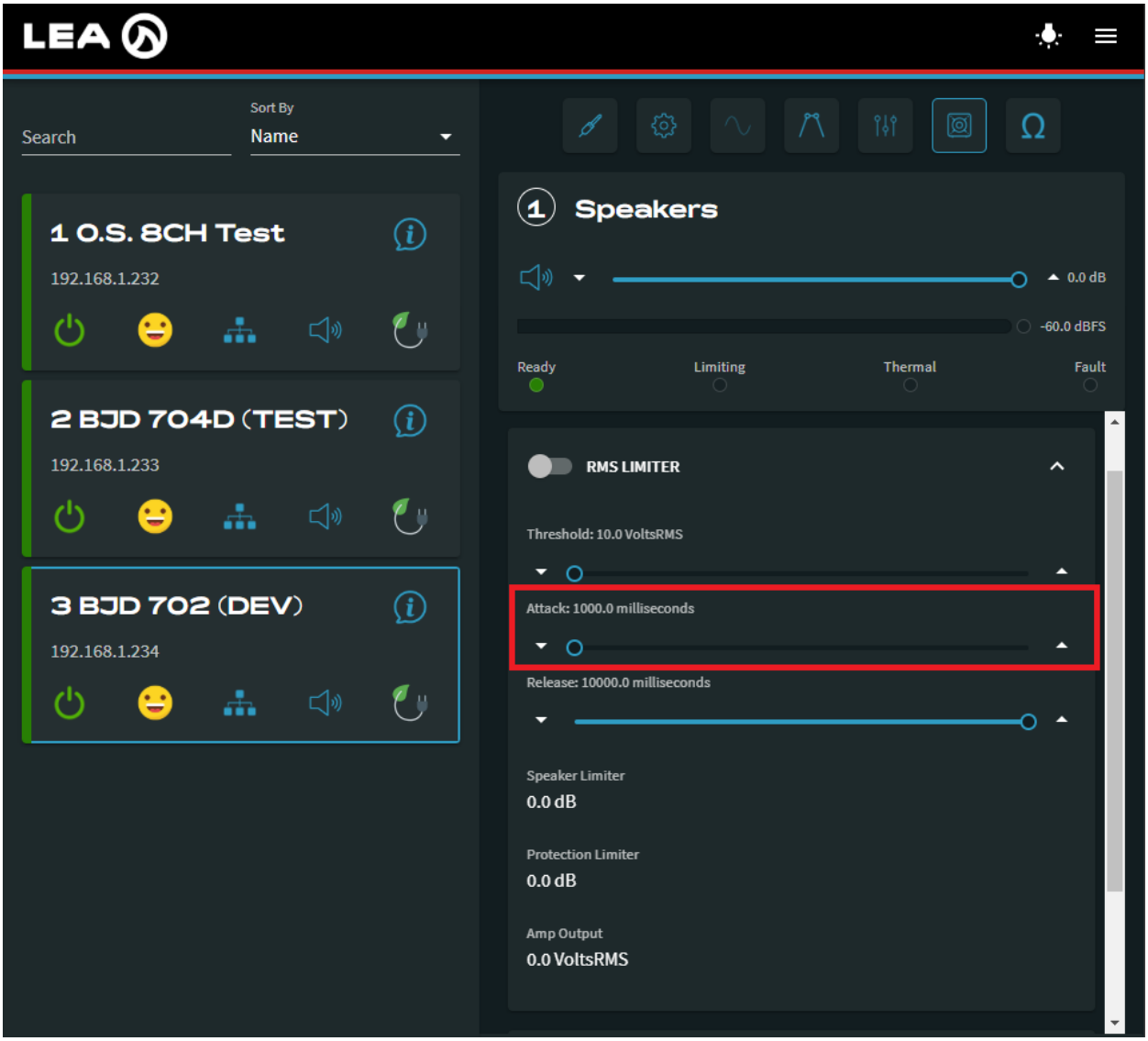
- This command set the RMS Limiter threshold to 70.0 V on channel 1



RMS Limiter Attack

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/rmsLimiter/attackTime
Values: 1000.0 through 10000.0
Example: set /amp/channels/1/rmsLimiter/attackTime 1000.0\n

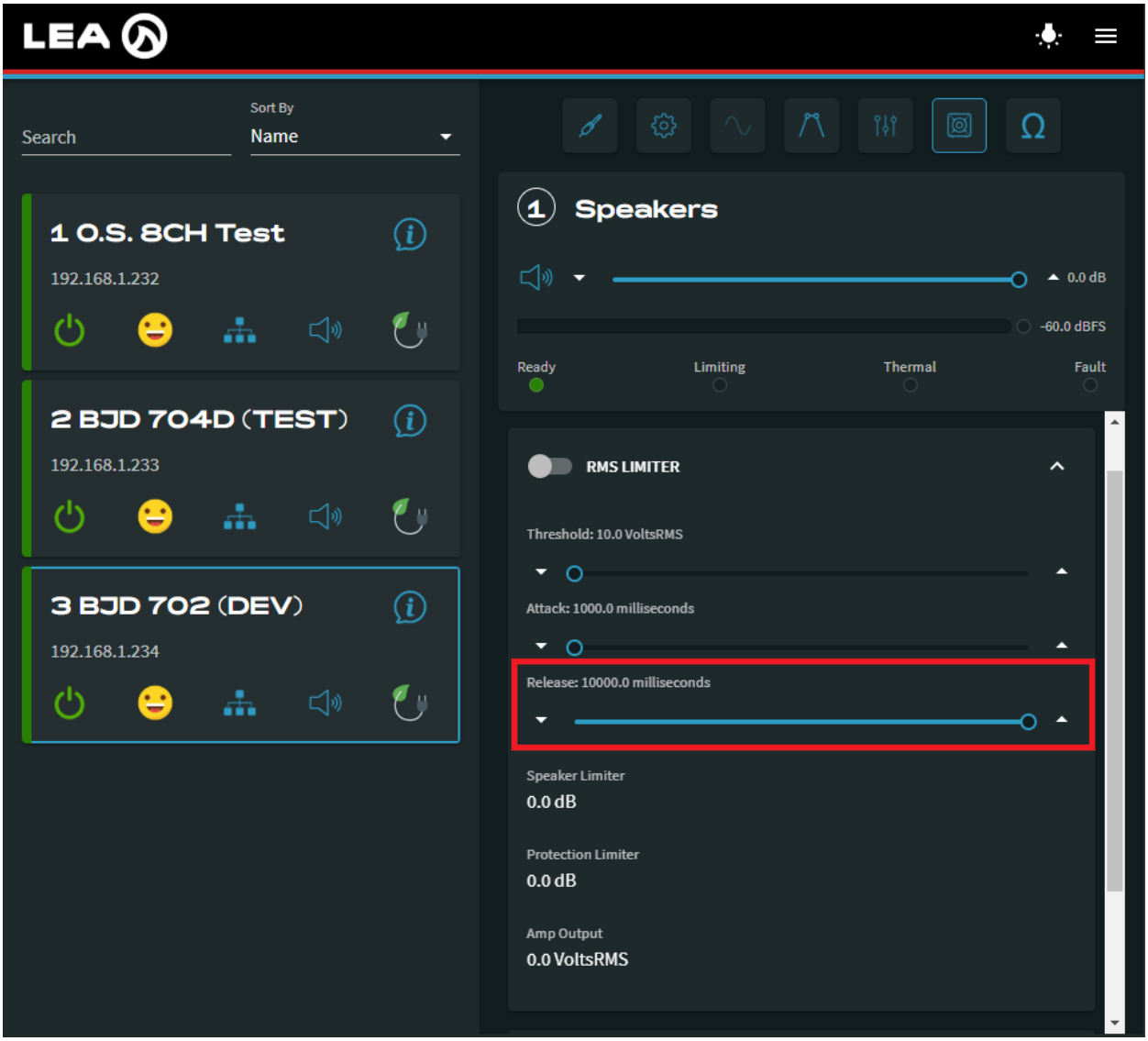
- This command set the RMS Limiter attack time to 1000.0 ms on channel 1



RMS Limiter Release

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/rmsLimiter/releaseTime
Values: 1000.0 through 10000.0
Example: set /amp/channels/1/rmsLimiter/releaseTime 10000.0\n

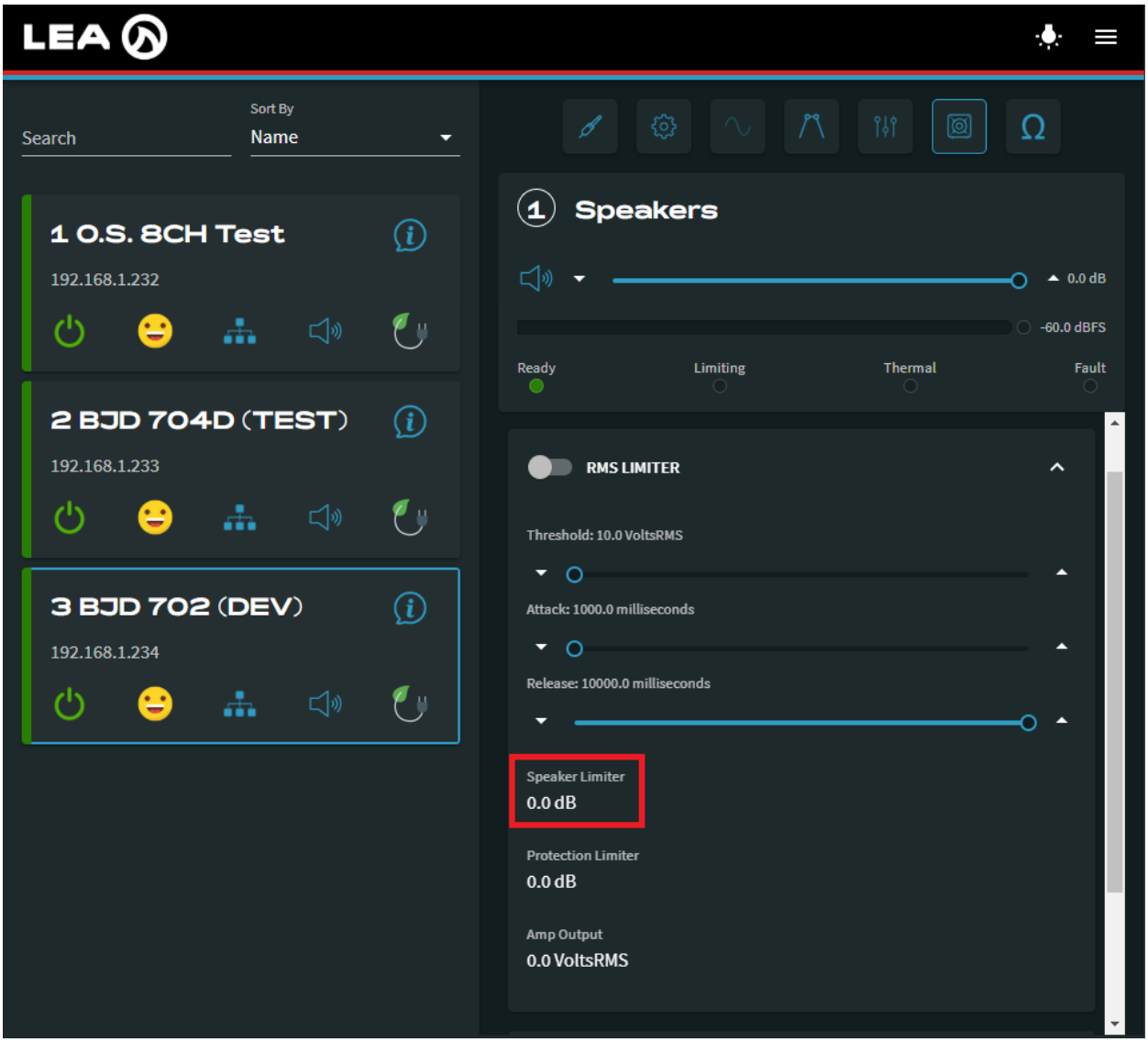
- This command set the RMS Limiter release time to 10000.0 ms on channel 1



RMS Limiter Speaker Limiter Reduction

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/rmsLimiter/gainReduction
Values: -80.0 through 0.0
Example: subscribe /amp/channels/1/rmsLimiter/gainReduction\n

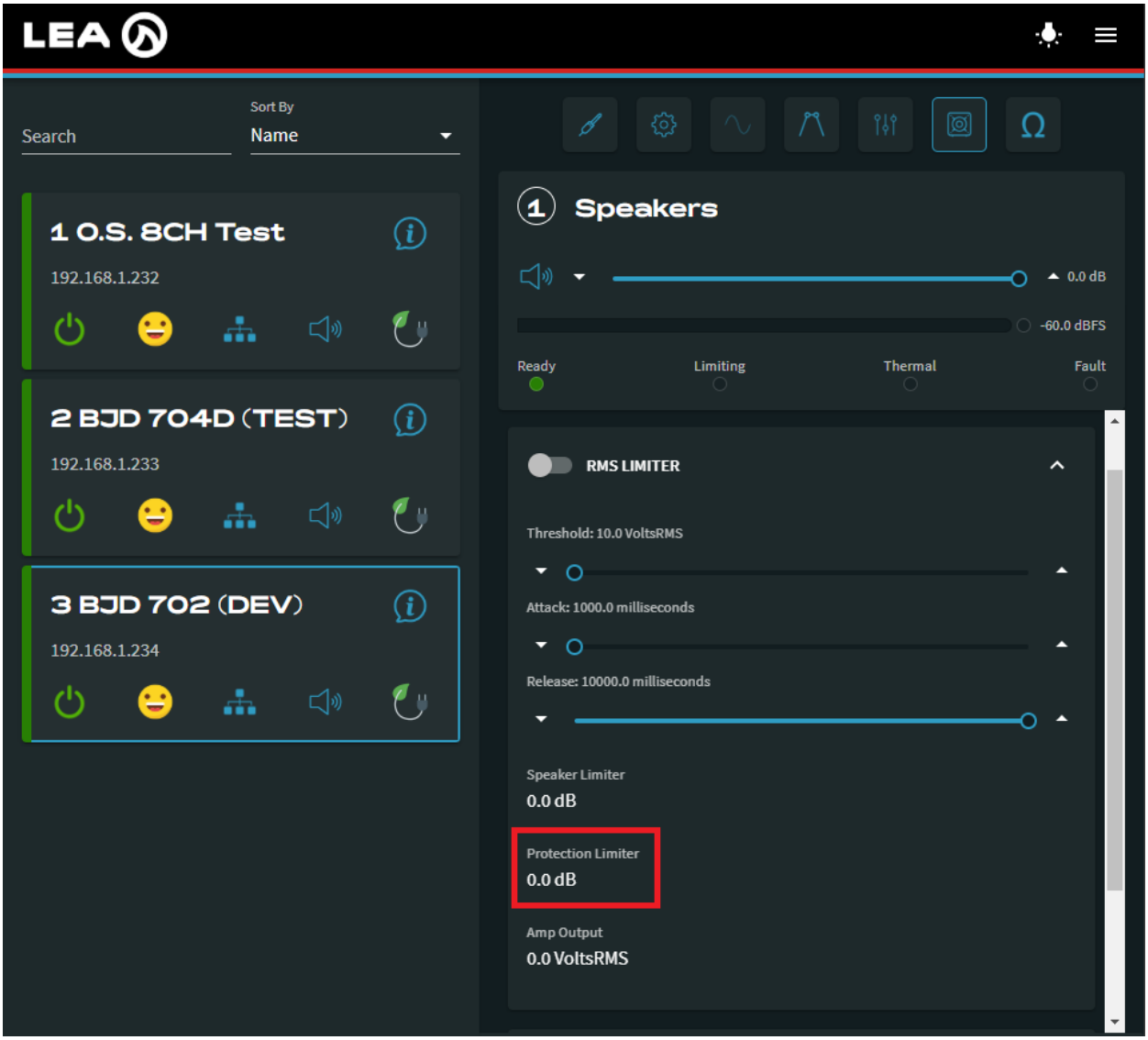
- This command subscribed to the RMS Limiter speaker limiter reduction sensor
- This is the active gain reduction from the user defined limiters



RMS Limiter Protection Limiter Reduction

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/rmsLimiter/totalGainReduction
Values: -80.0 through 0.0
Example: subscribe /amp/channels/1/rmsLimiter/totalGainReduction\n

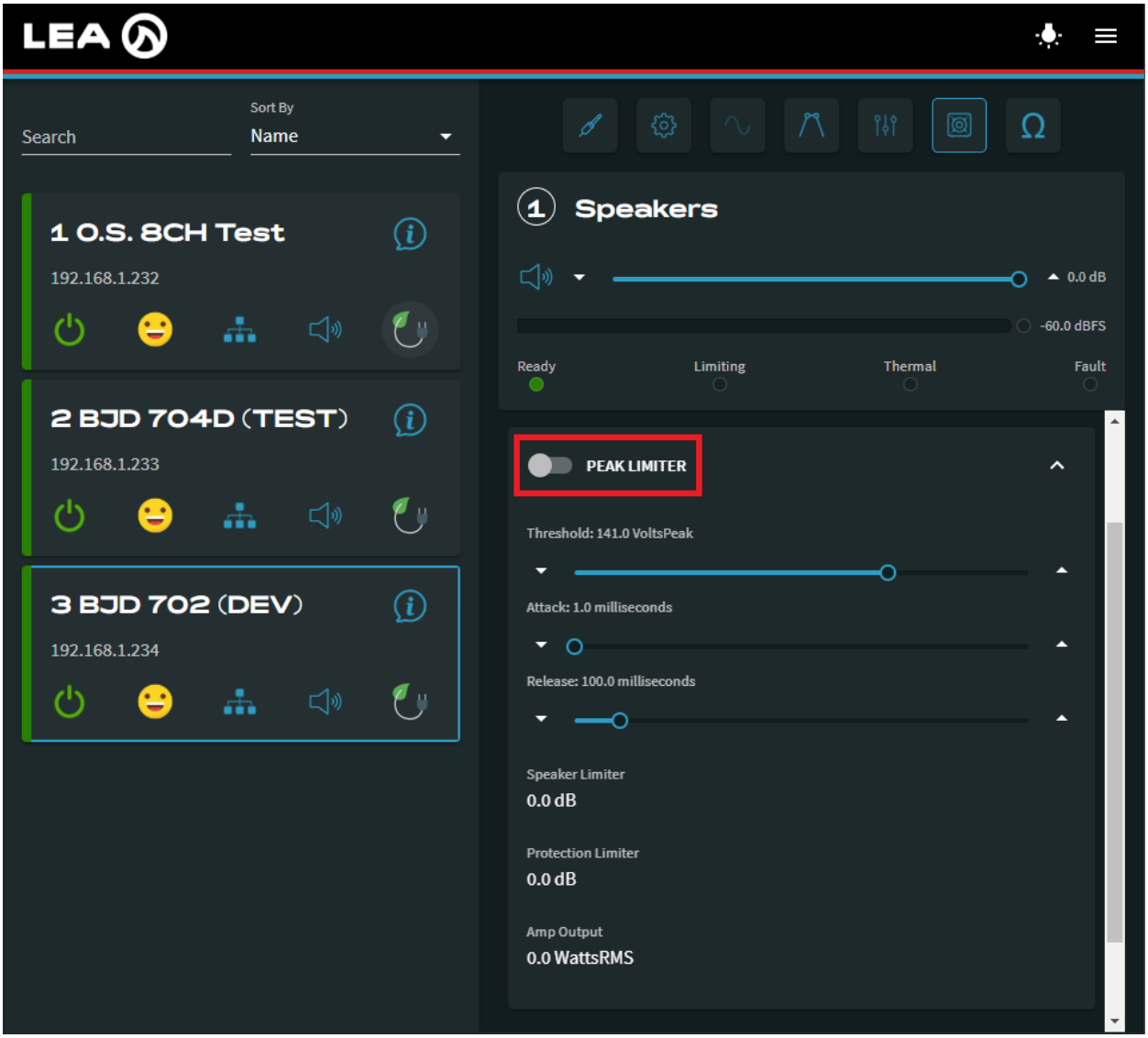
- This command subscribed to the RMS Limiter protection limiter reduction sensor
- This is the total active gain reduction applied to the amplifier from both the user defined limiters and internal amplifier protection limiters



Peak Limiter Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/peakLimiter/enable
Values: "true", "false"
Example: set /amp/channels/1/peakLimiter/enable "false"\n

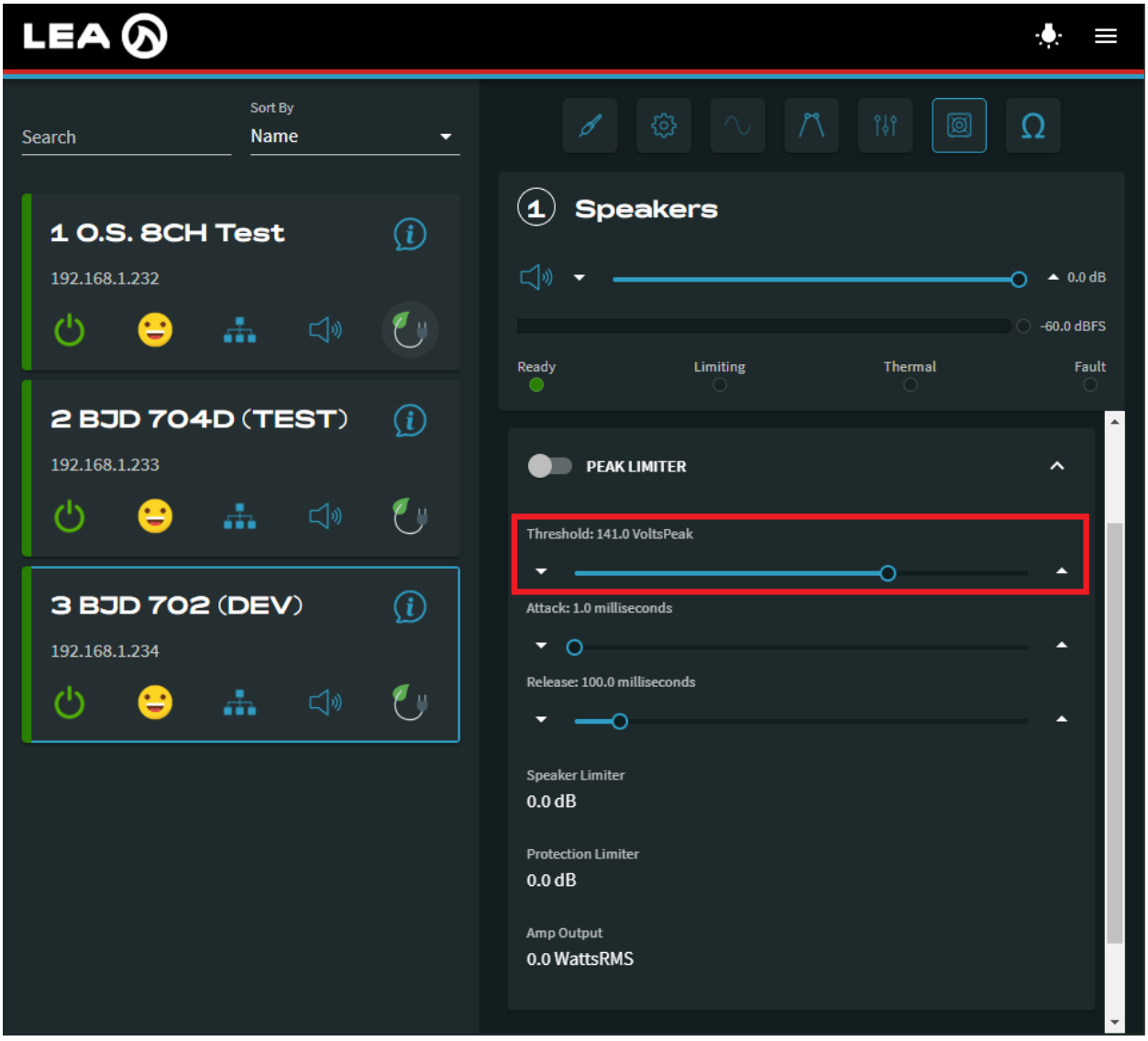
- This command disabled the Peak Limiter on channel 1



Peak Limiter Threshold

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/peakLimiter/threshold
Values: 14.0 through 198.0
Example: set /amp/channels/1/peakLimiter/threshold 141.0\n

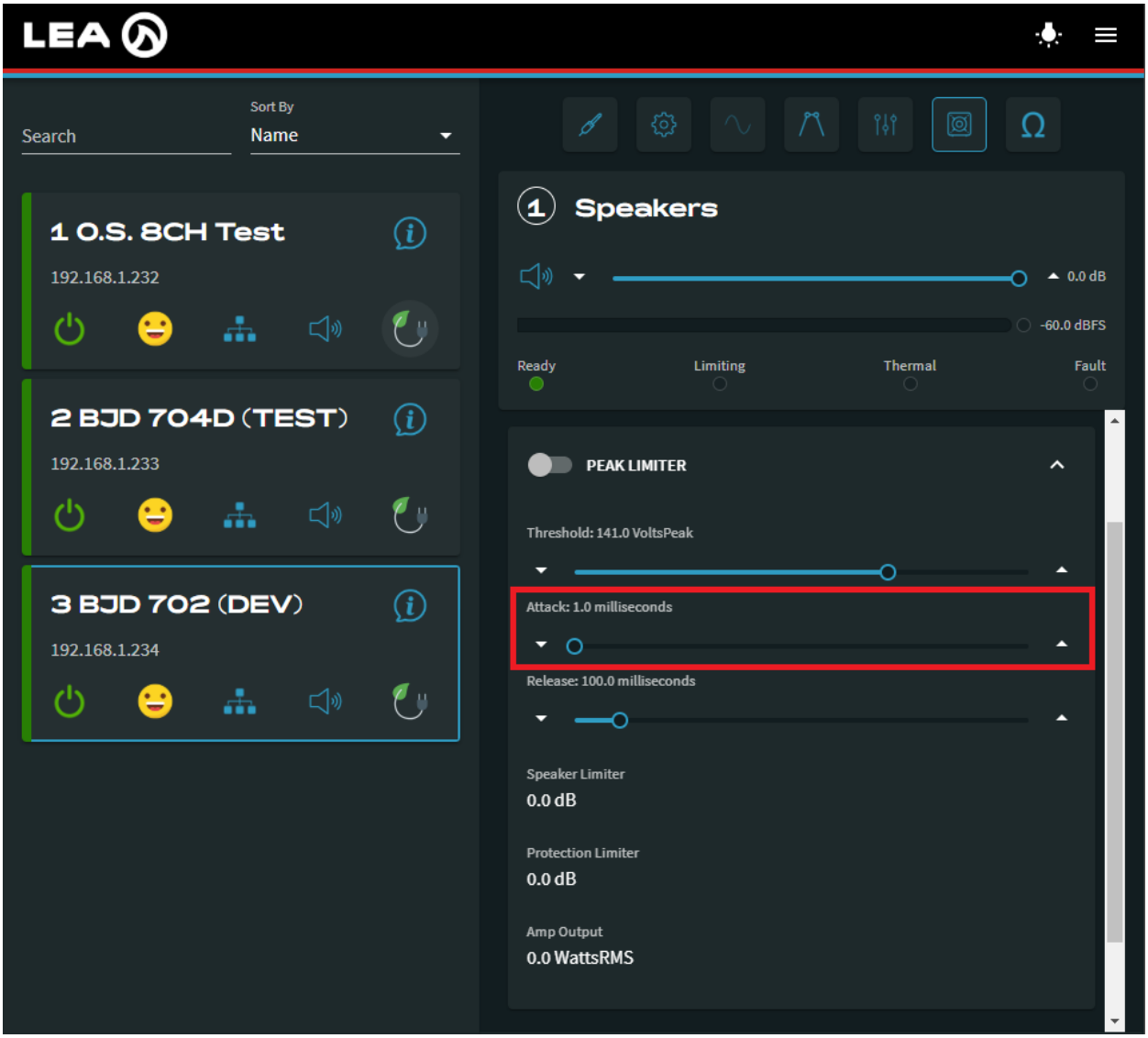
- This command set the Peak Limiter threshold to 141.0 V on channel 1



Peak Limiter Attack

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/peakLimiter/attackTime
Values: 1.0 through 1000.0
Example: set /amp/channels/1/peakLimiter/attackTime 10.0\n

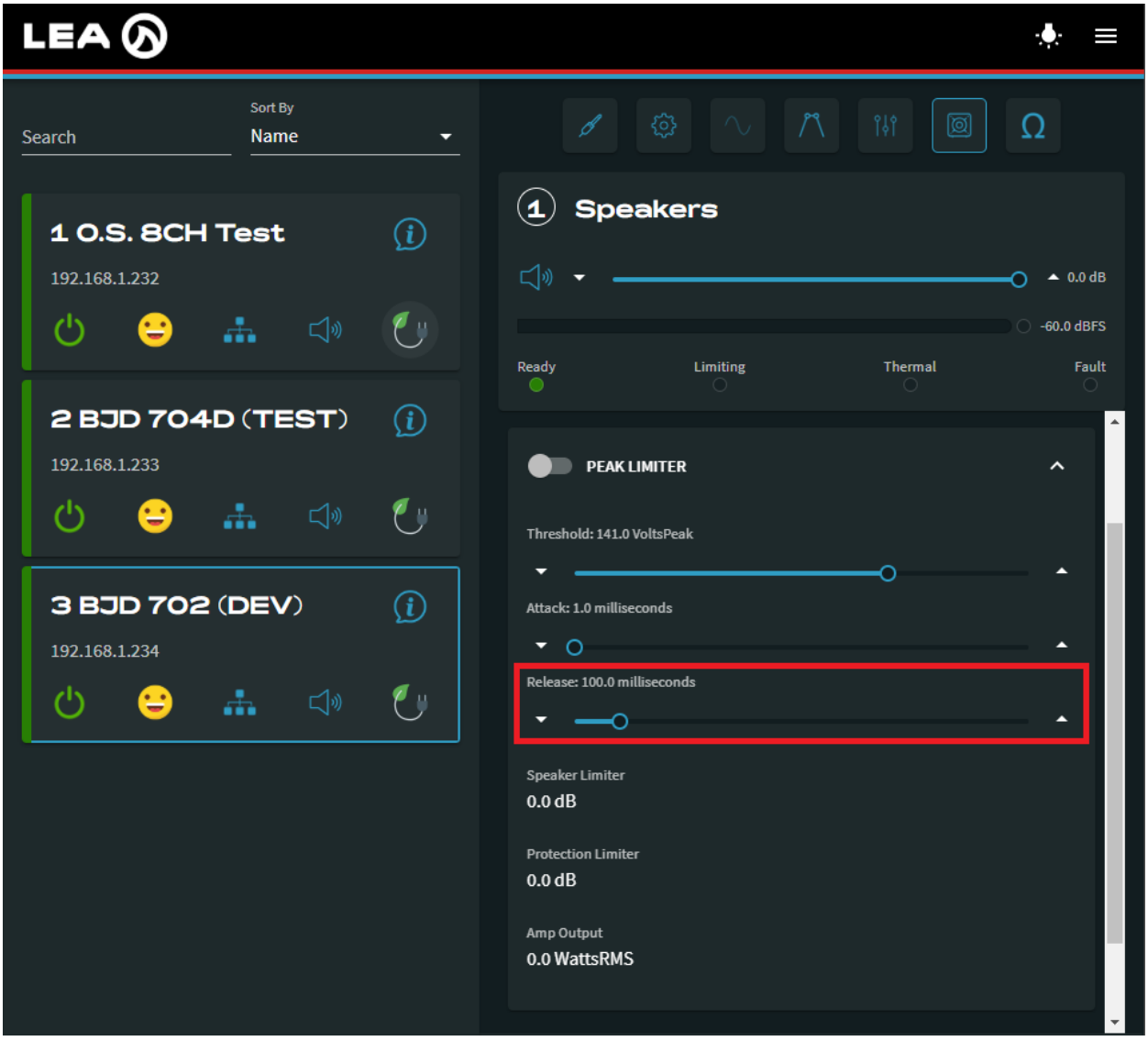
- This command set the Peak Limiter attack time to 10.0 ms on channel 1



Peak Limiter Release

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/peakLimiter/releaseTime
Values: 1.0 through 1000.0
Example: set /amp/channels/1/peakLimiter/releaseTime 100.0\n

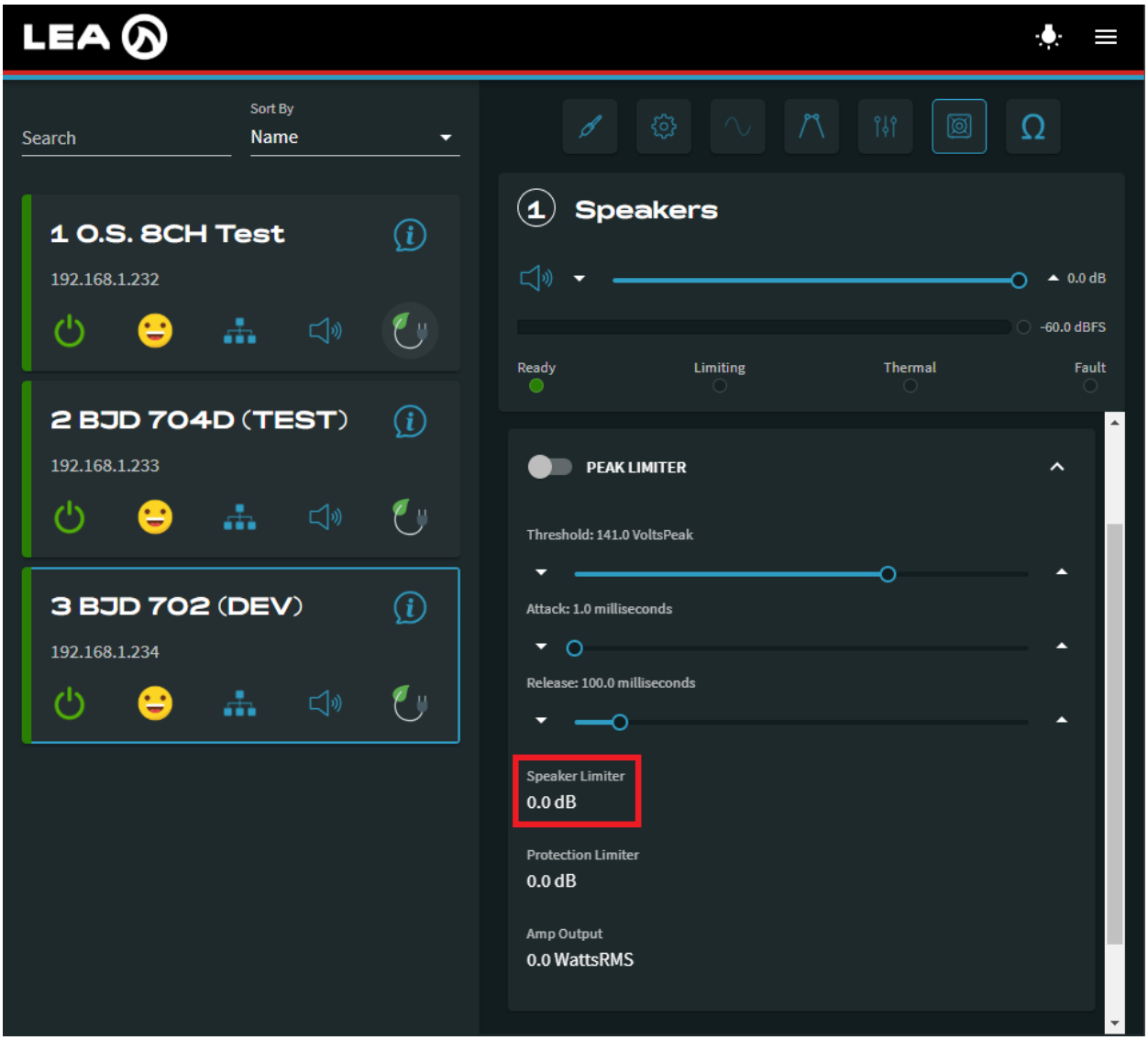
- This command set the Peak Limiter release time to 100.0 ms on channel 1



Peak Limiter Speaker Limiter Reduction

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/peakLimiter/gainReduction
Values: -80.0 through 0.0
Example: subscribe /amp/channels/1/peakLimiter/gainReduction\n

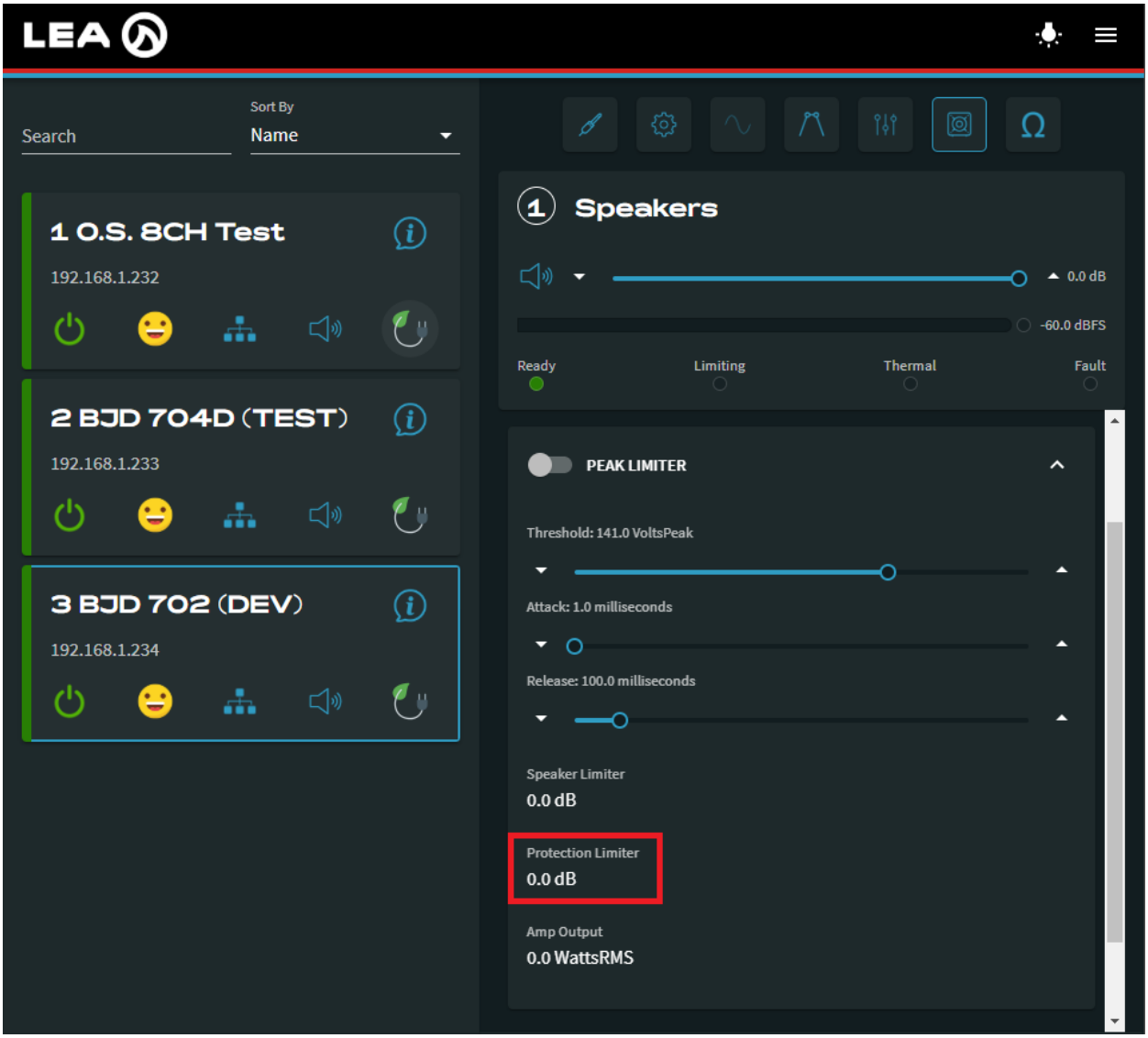
- This command subscribed to the Peak Limiter speaker limiter reduction sensor
- This is the active gain reduction from the user defined limiters



Peak Limiter Protection Limiter Reduction

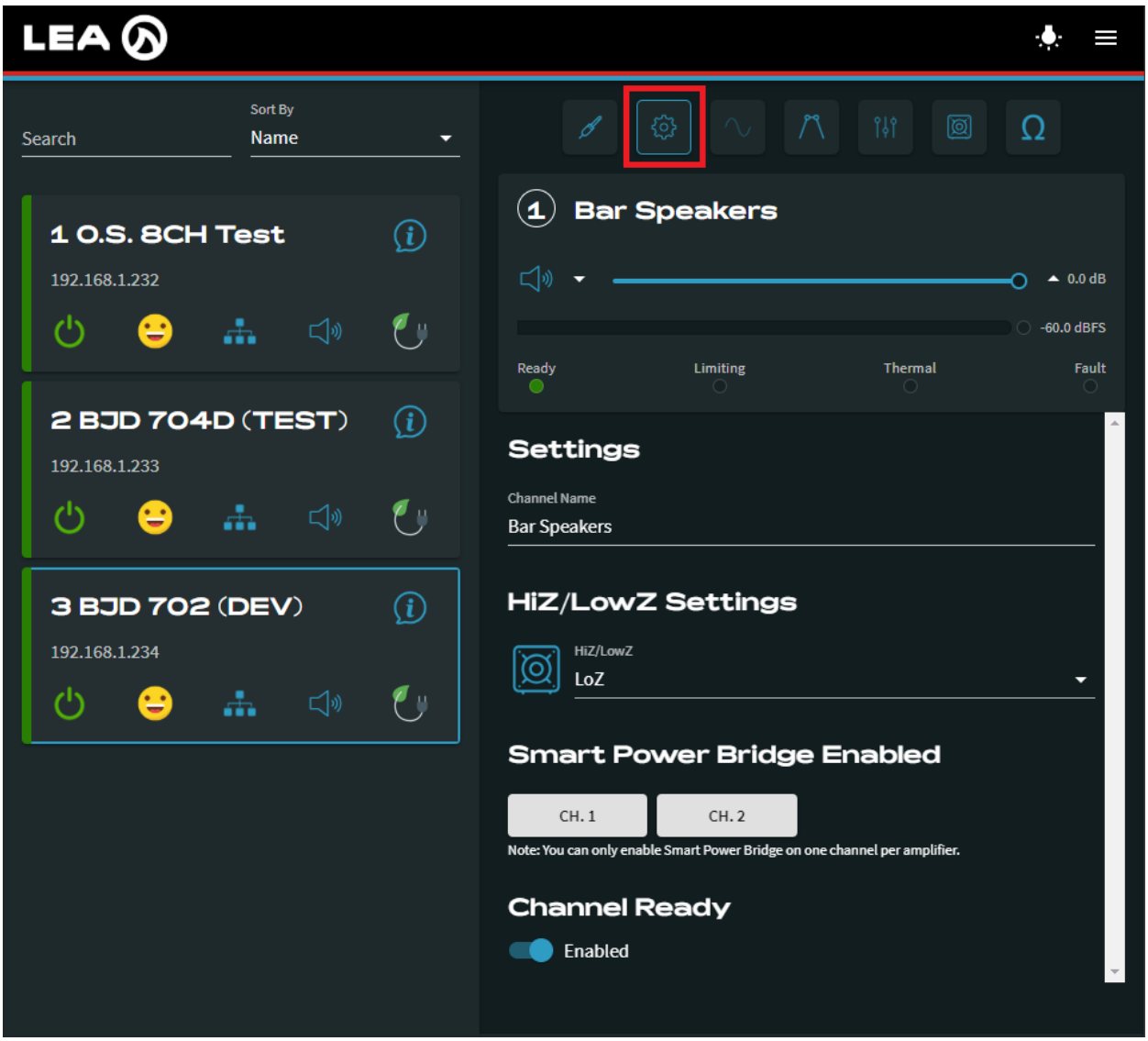
Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/peakLimiter/totalGainReduction
Values: -80.0 through 0.0
Example: subscribe /amp/channels/1/peakLimiter/totalGainReduction\n

- This command subscribed to the Peak Limiter protection limiter reduction sensor
- This is the total active gain reduction applied to the amplifier from both the user defined limiters and internal amplifier protection limiters



Amplifier Channels Output

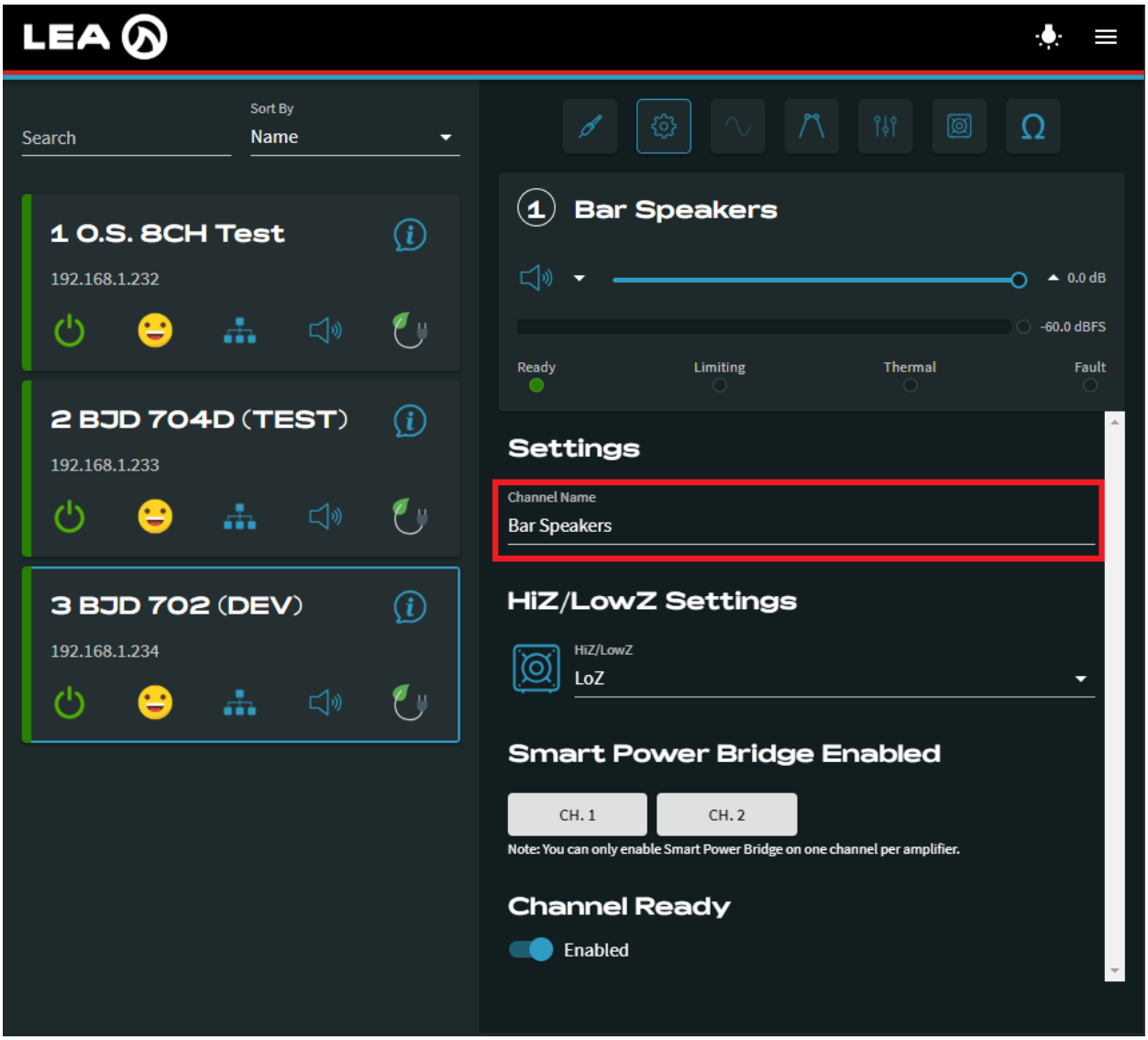
Click on this button to navigate to the General Channel Settings of the DSP relevant to this API section



Output Channel Name

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/output/name
Values: any text up to 64 characters
Example: get /amp/channels/1/output/name\n

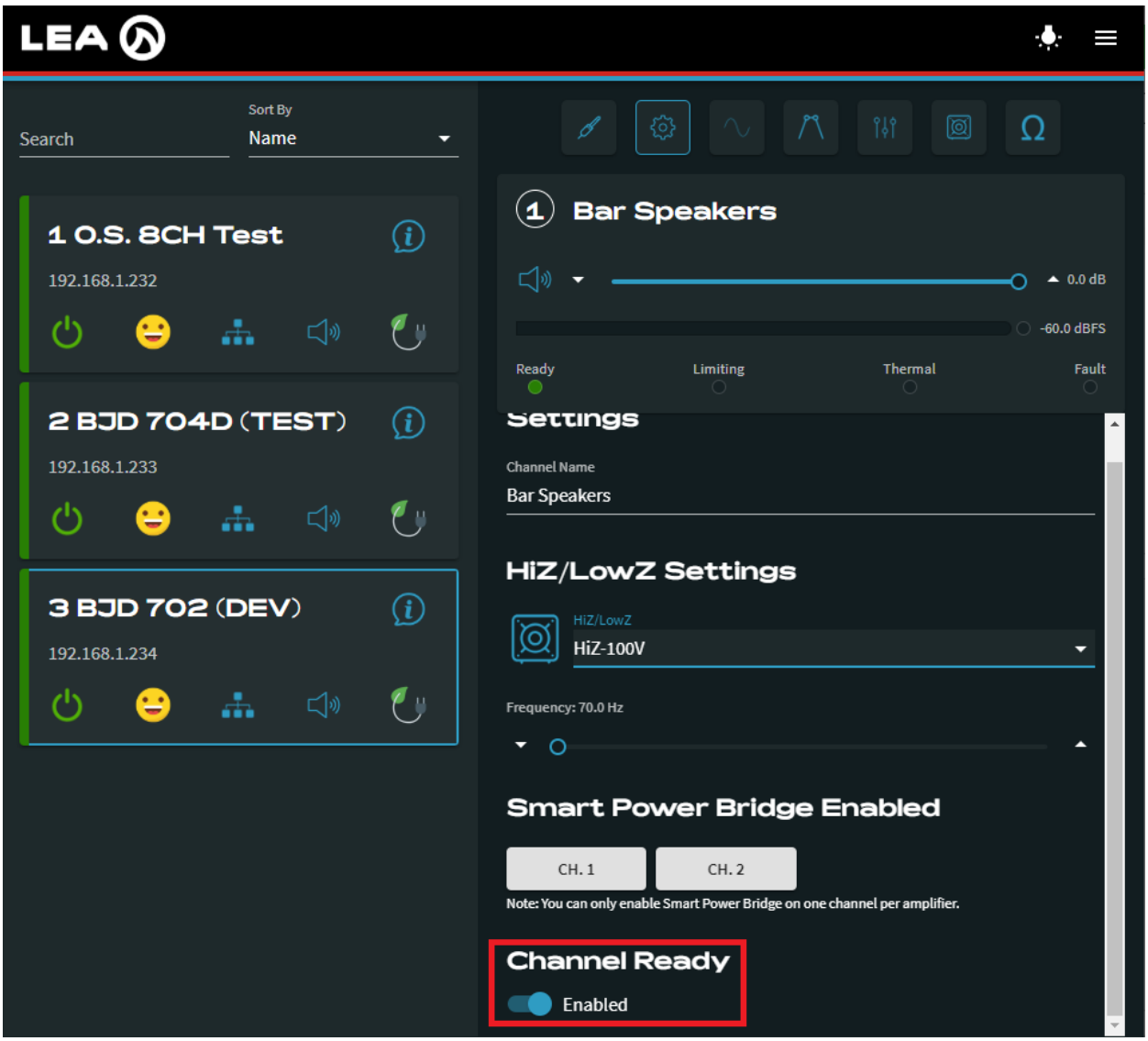
- Response: amp/channels/1/output/name "Bar Speakers"\n
- This command asked for the output channel name and got the response Bar Speakers



Output Channel Ready Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/output/enable
Values: "true", "false"
Example: set /amp/channels/1/output/enable "true"\n

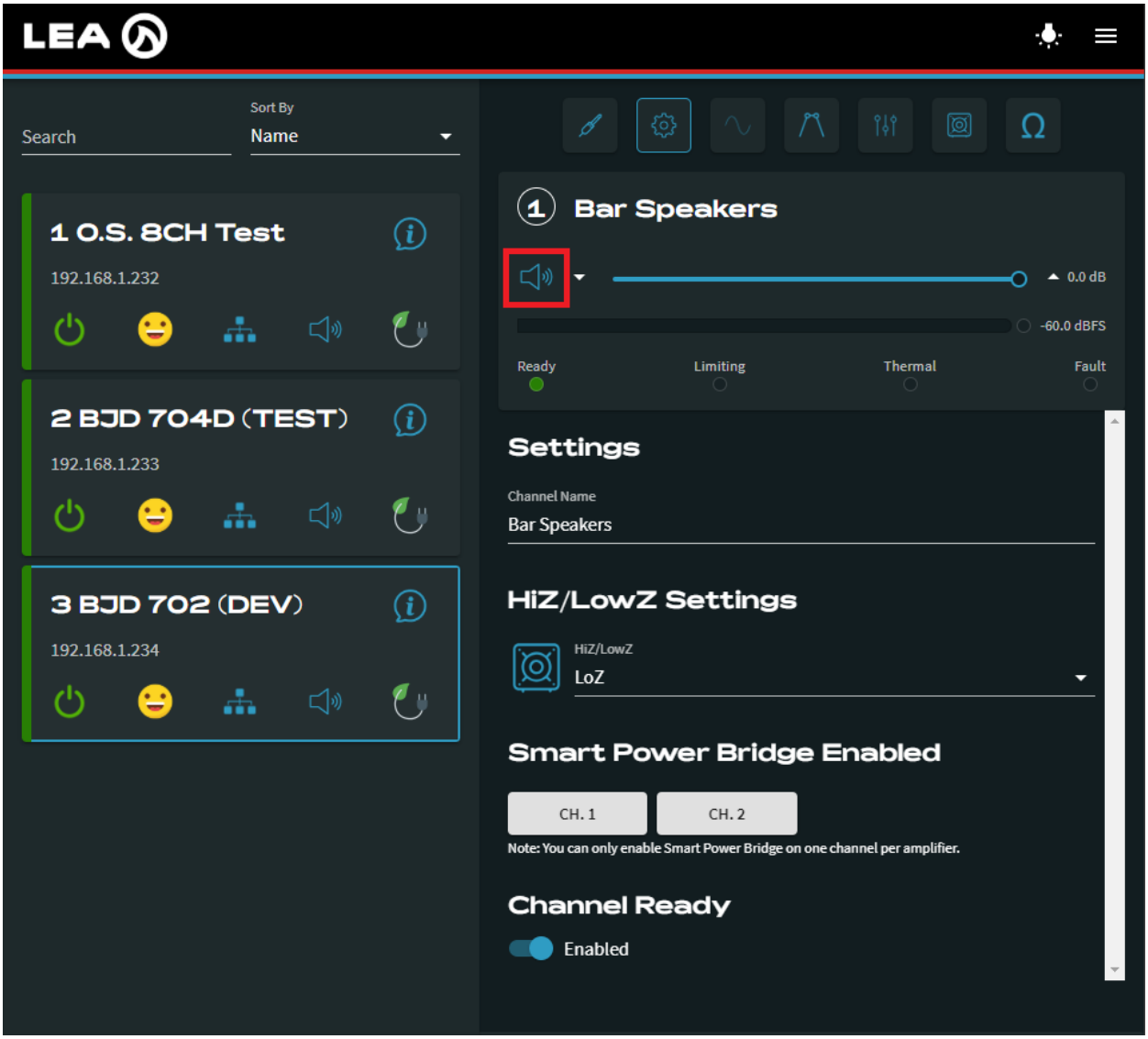
- This command enabled the channel ready on channel 1



Output Channel Mute

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/output/mute
Values: "true", "false"
Example: set /amp/channels/1/output/mute "false"\n

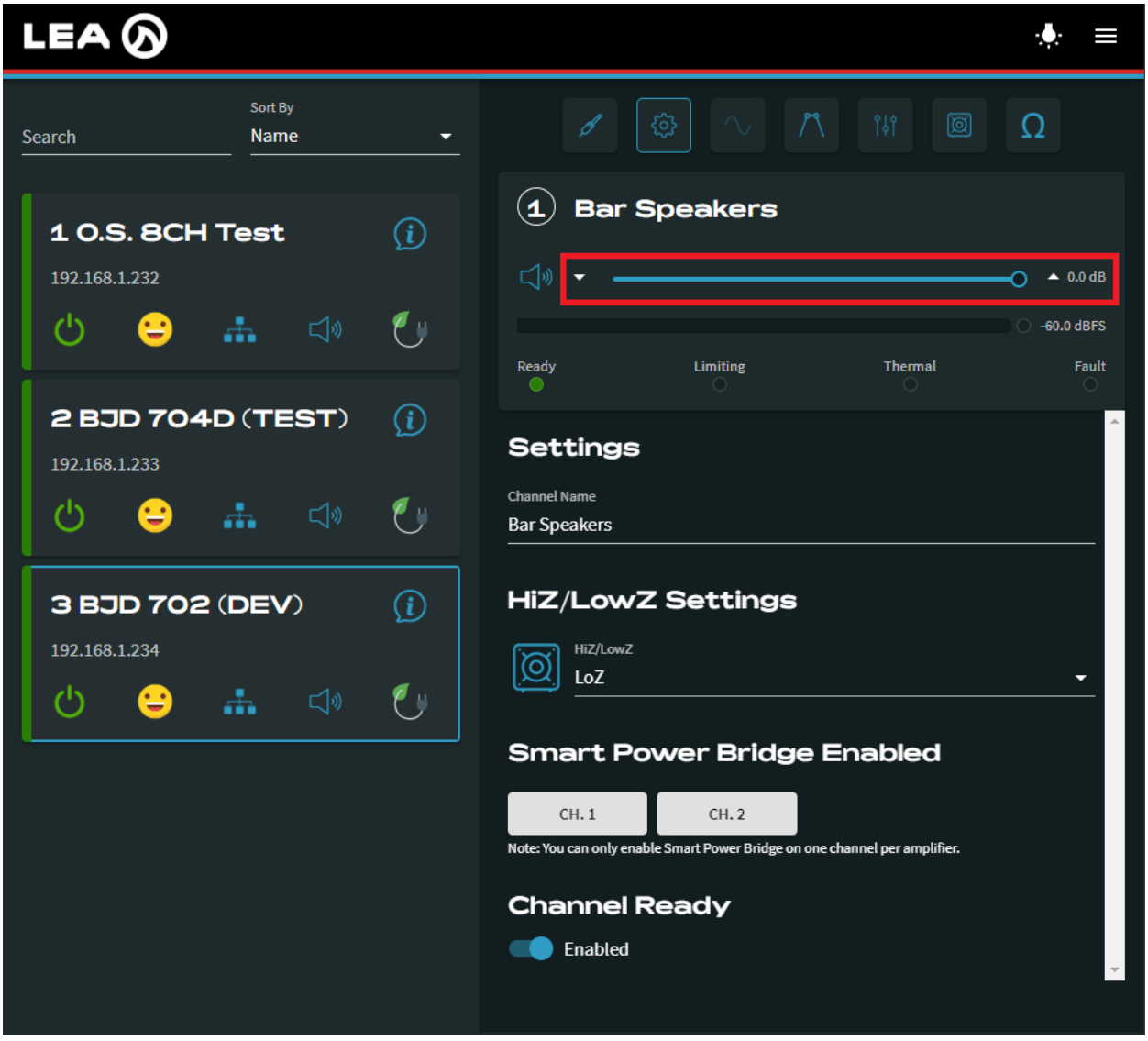
- This command un-muted the output on channel 1



Output Channel Gain Attenuation Fader

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/output/fader
Values: -80.0 through 0.0
Example: set /amp/channels/1/output/fader 0.0\n

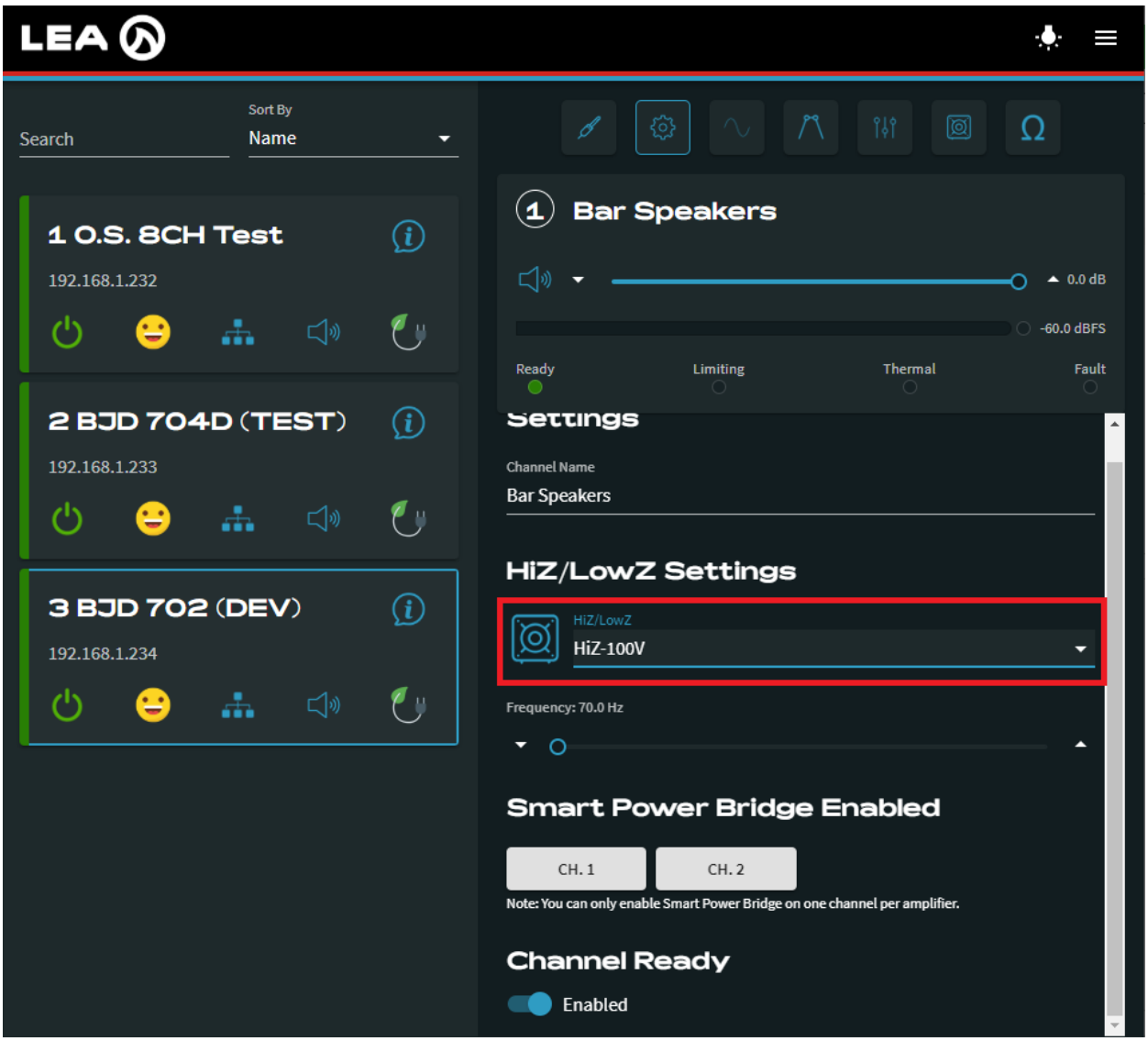
- This command set the output attenuation to 0.0 dB on channel 1



Output Channel Hi-Z Low-Z Mode

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/output/hiZLoZ
Values: "HiZ-70V", "HiZ-100V", "LoZ"
Example: set /amp/channels/1/output/hiZLoZ "HiZ-100V"\n

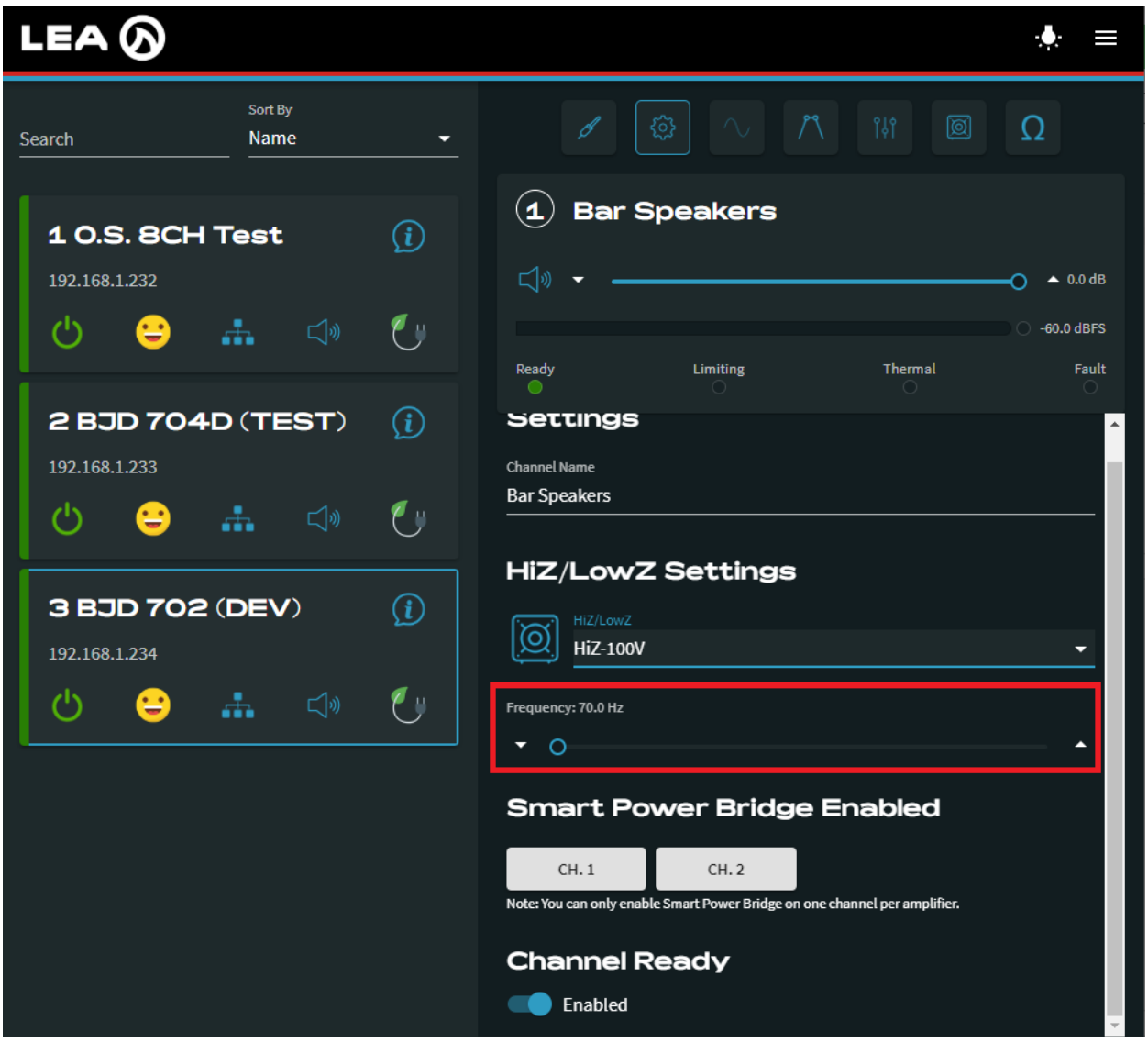
- This command set channel 1 to Hi-Z 100 V mode



Output Channel Hi-Z Mode High Pass Frequency

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/output/hiZHpfFrequency
Values: 35 through 5000
Example: set /amp/channels/1/output/hiZHpfFrequency 70\n

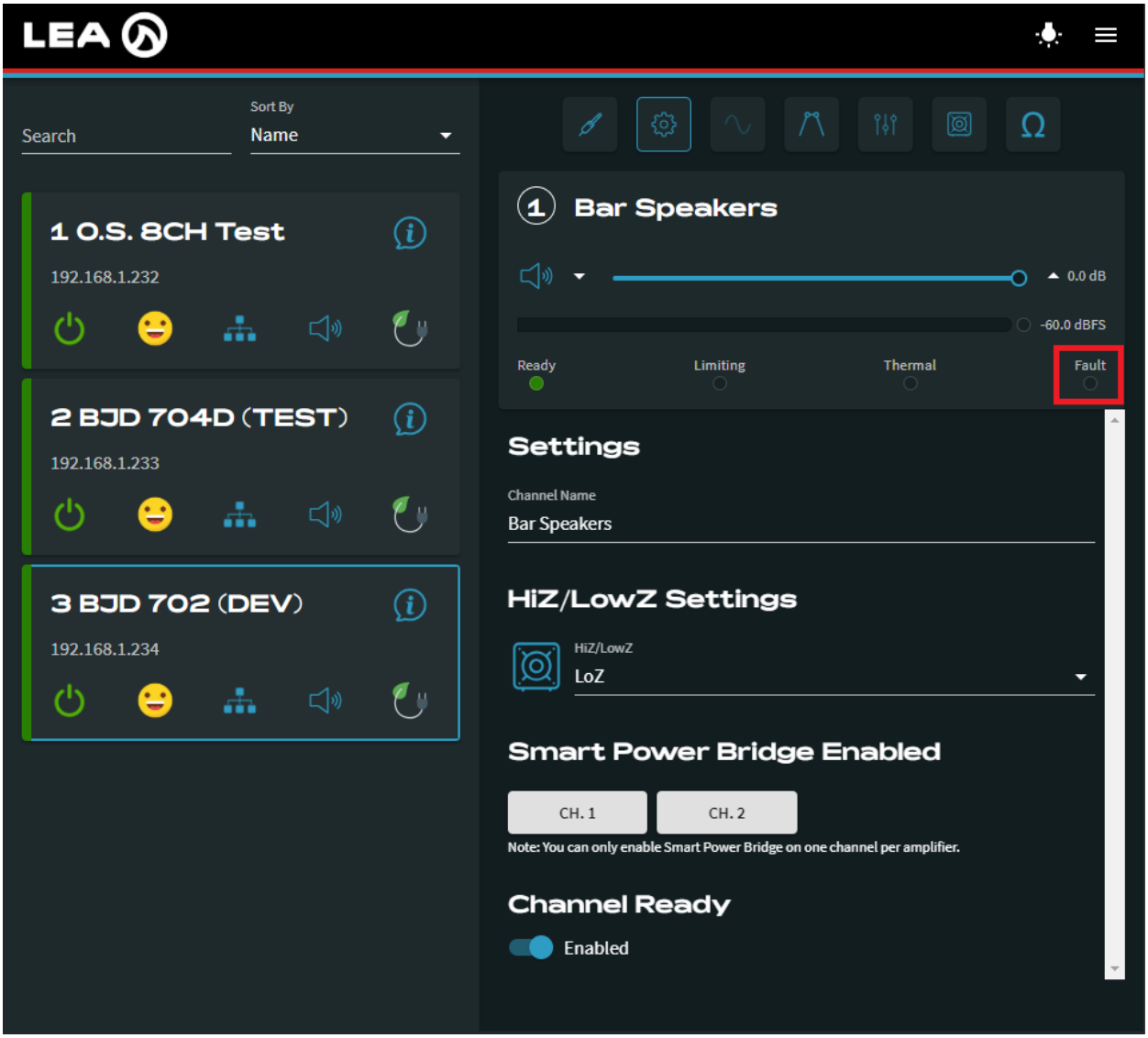
- This command set Hi-Z High Pass Filter to 70 Hz on channel 1



Output Channel Fault

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/output/fault
Values: "true", "false"
Example: get /amp/channels/1/output/fault\n

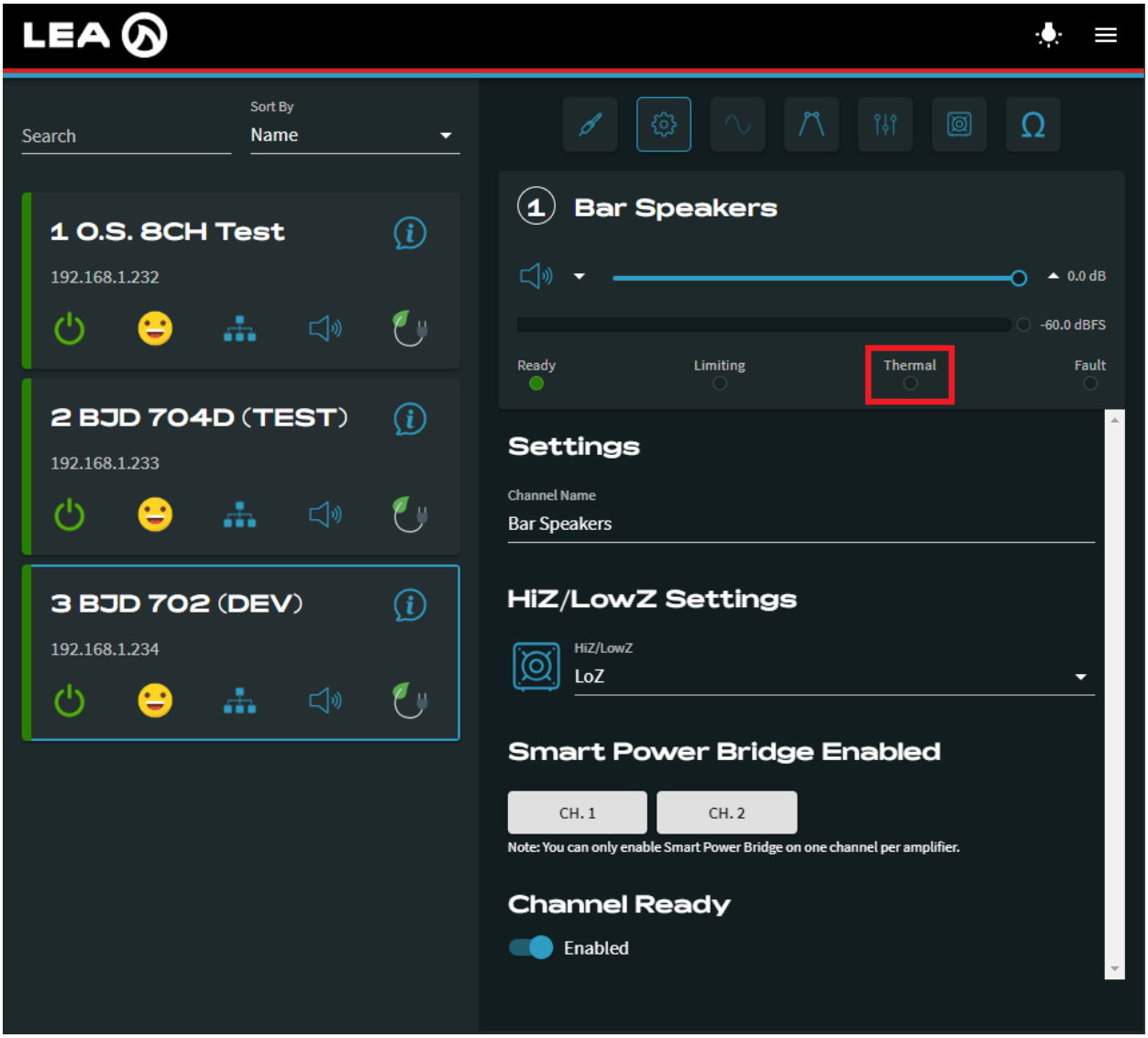
- Response: /amp/channels/1/output/fault false\n
- This command asked for the fault status on channel 1 and got the response False, meaning there is no fault on the channel



Output Channel Thermal Fault

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/output/thermal
Values: "true", "false"
Example: get /amp/channels/1/output/thermal\n

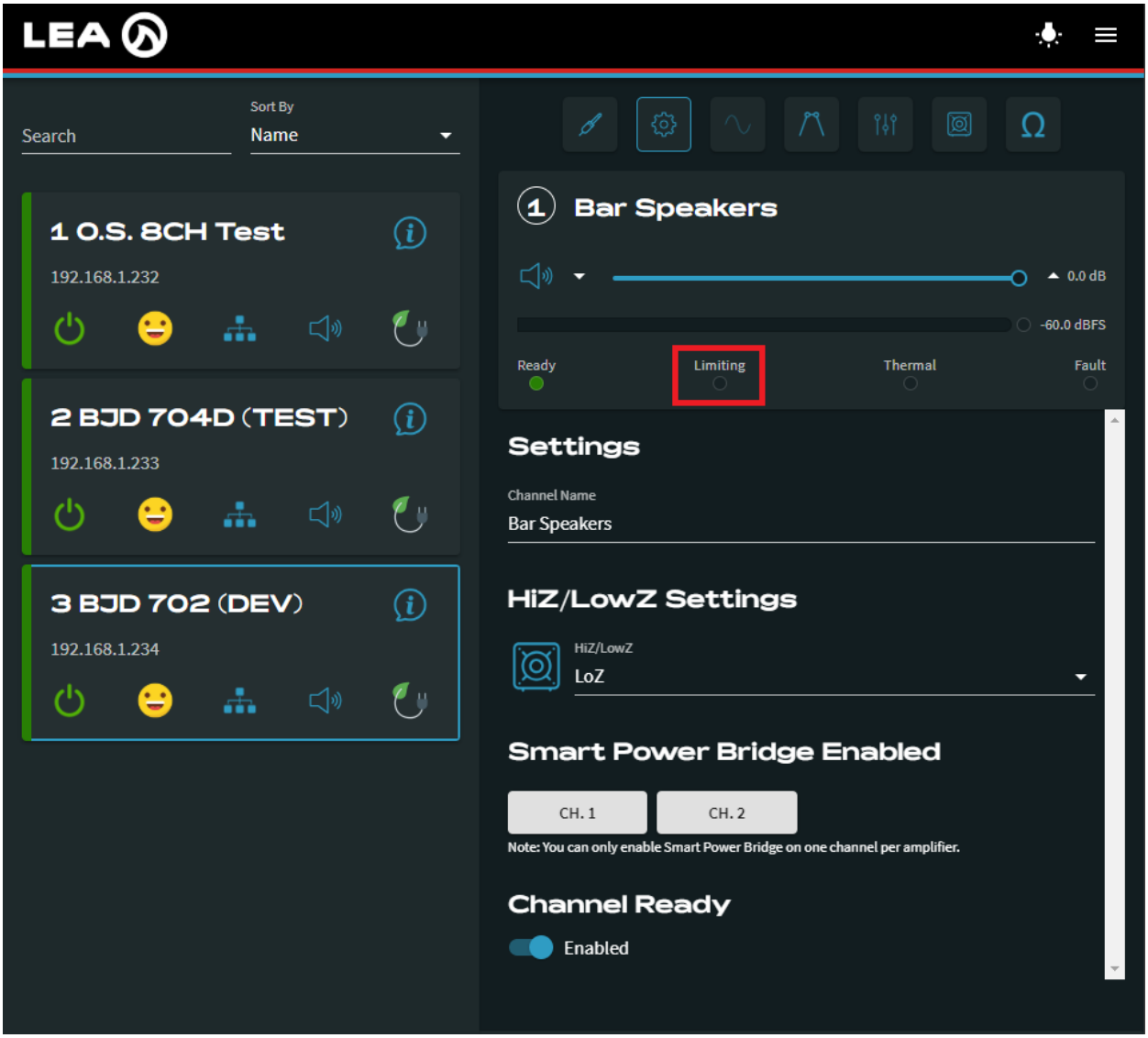
- Response: /amp/channels/1/output/thermal false\n
- This command asked for the thermal fault status on channel 1 and got the response False, meaning there is no thermal fault on the channel



Output Channel Limiting

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/output/limiting
Values: "true", "false"
Example: get /amp/channels/1/output/limiting\n

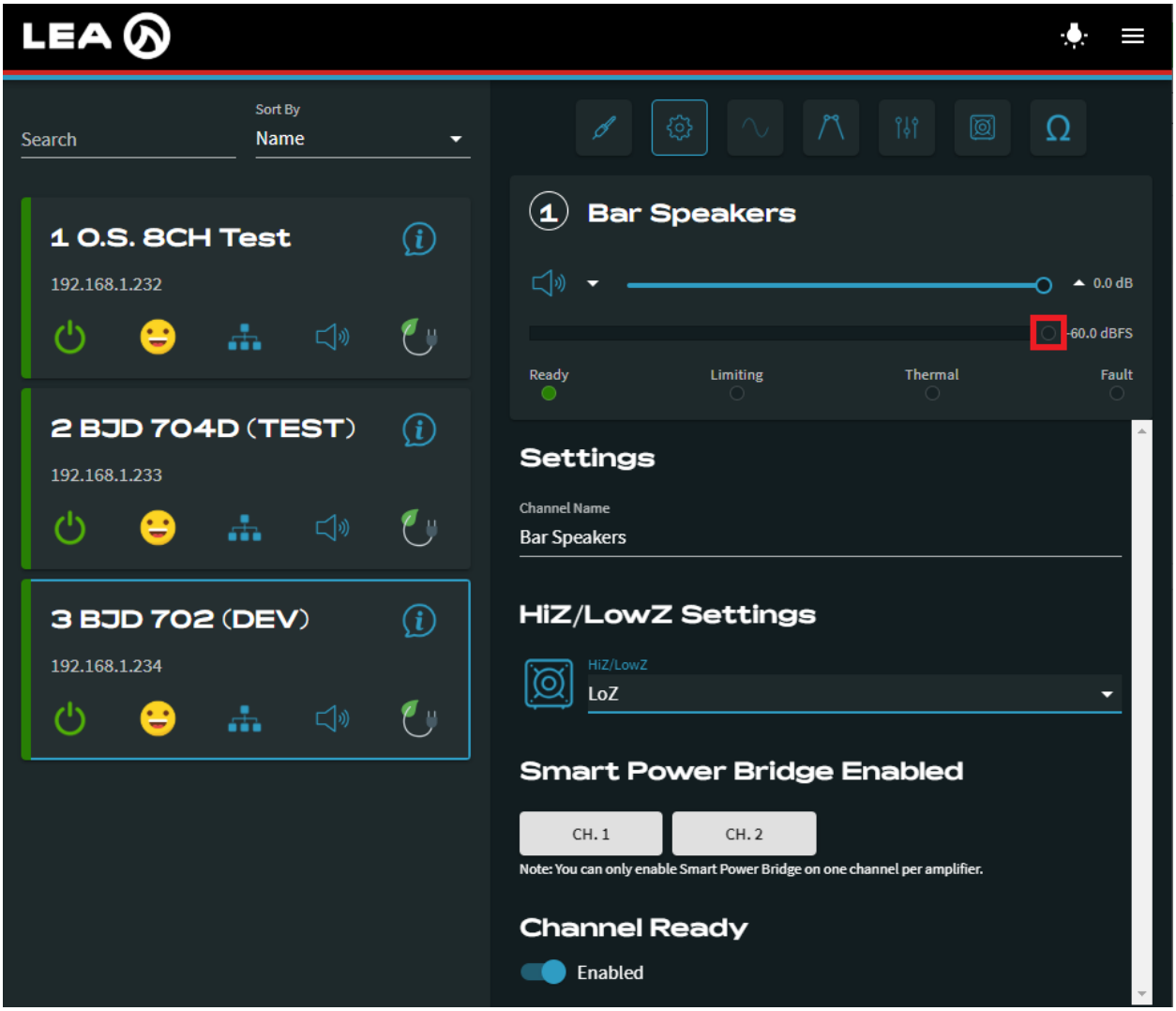
- Response: /amp/channels/1/output/thermal false\n
- This command asked for the limiting status on channel 1 and got the response False, meaning there is no limiting active on the channel



Output Channel Clip

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/output/clip
Values: "true", "false"
Example: get /amp/channels/1/output/clip\n

- Response: /amp/channels/1/output/clip false\n
- This command asked for the output clip status on channel 1 and got the response False, meaning the output is not being clipped on the channel



Output Channel Signal Presence

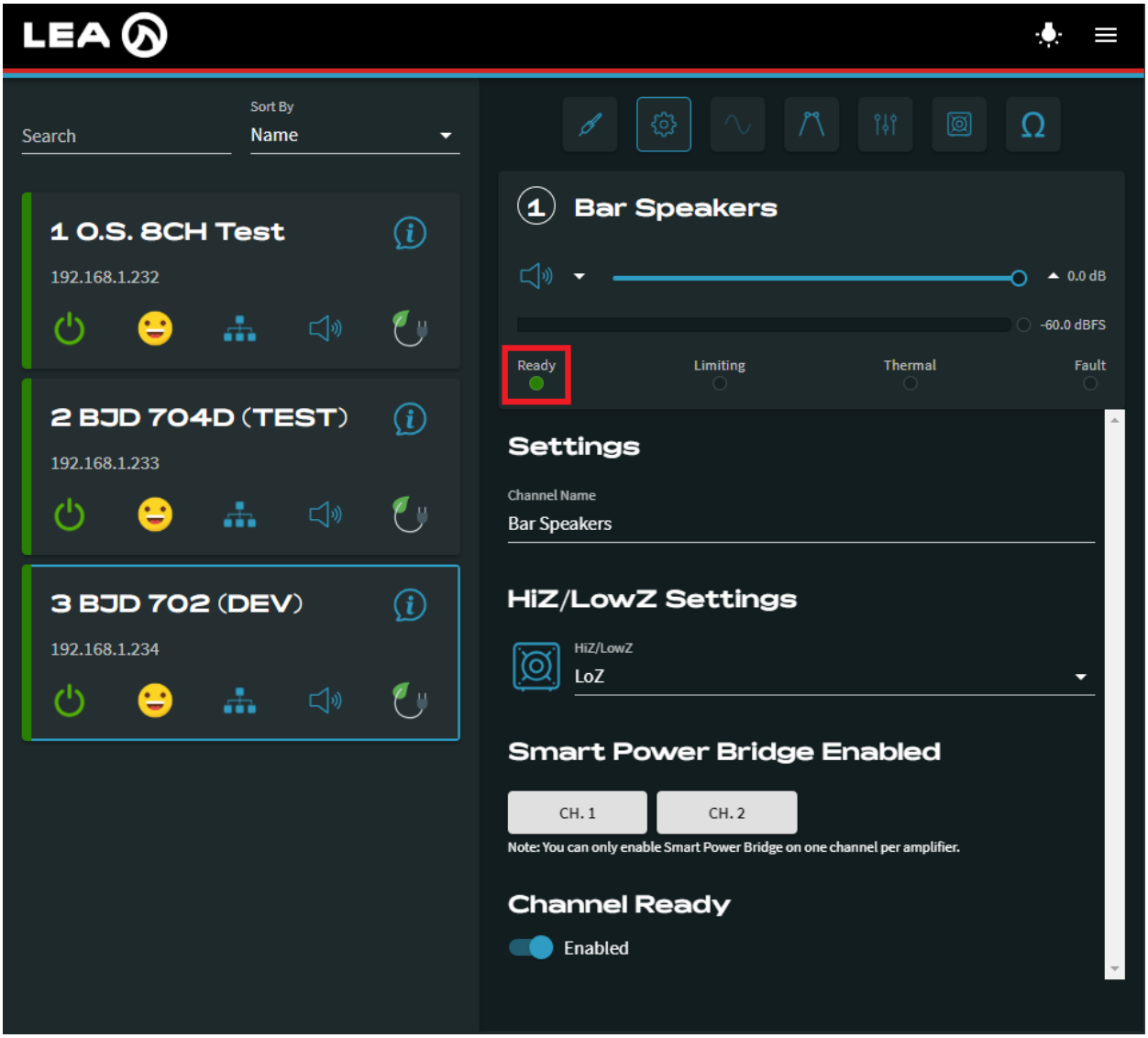
Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/output/signalDetect
Values: "true", "false"
Example: get /amp/channels/1/output/signalDetect\n

- Response: /amp/channels/1/output/signalDetect false\n
- This command asked for the output signal presence status on channel 1 and got the response False, meaning there is not output signal on the channel

Output Channel Ready Indicator

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/output/ready
Values: "true", "false"
Example: get /amp/channels/1/output/ready\n

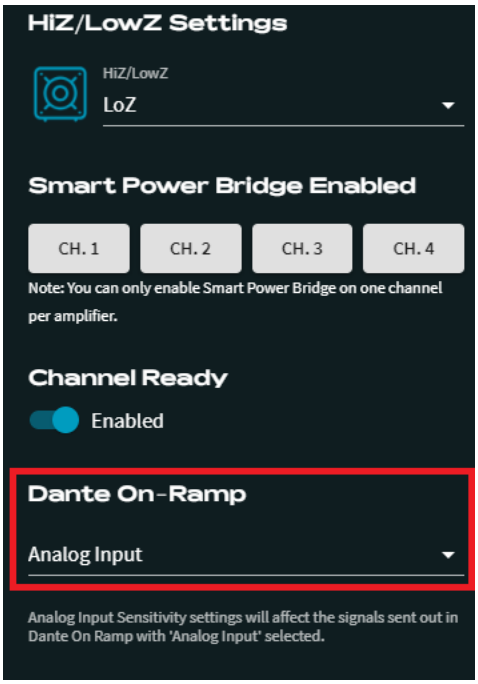
- Response: /amp/channels/1/output/ready true\n
- This command asked for the output ready status on channel 1 and got the response True, meaning the output channel is ready to pass audio



Dante On Ramp

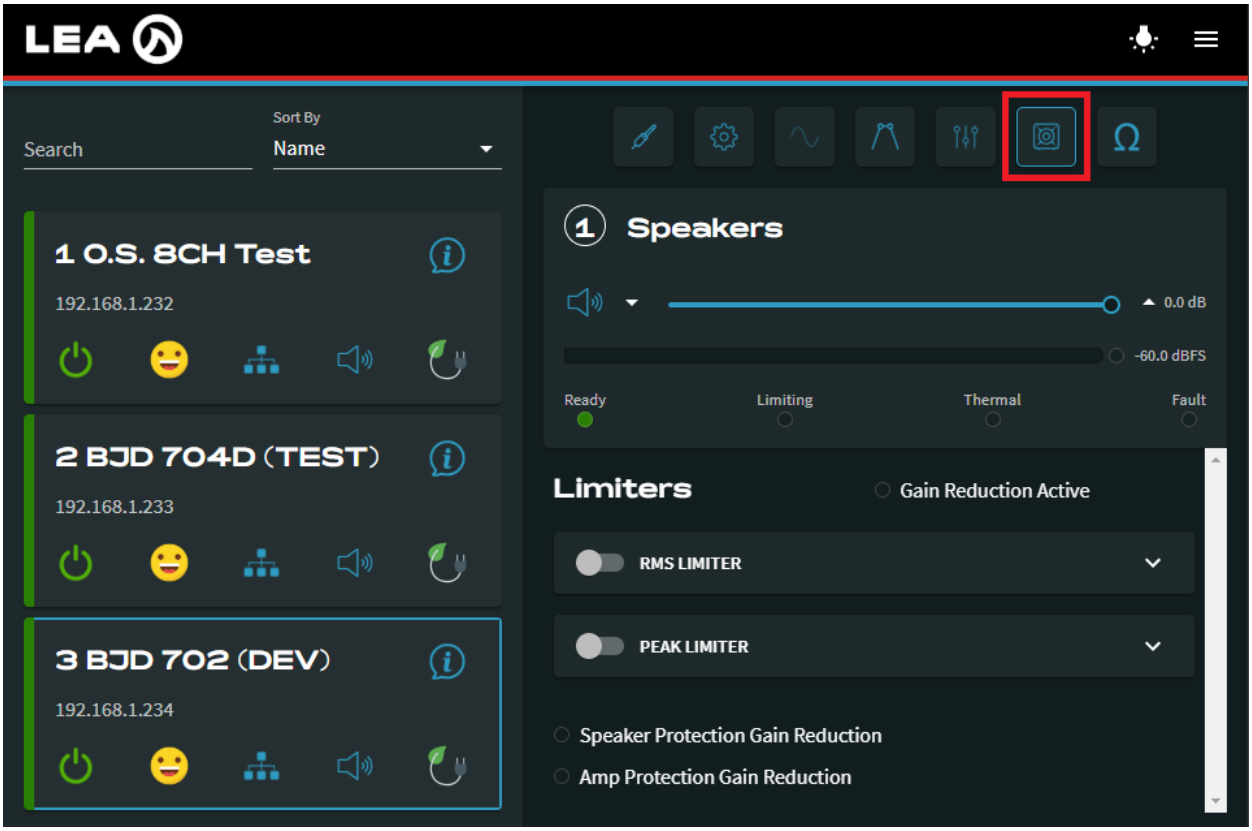
Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/inputSelector/danteOnRamp
Values: "Analog Input", "Post Crossover", "Amp Output Monitor", "Amp Imon"
Example: set /amp/channels/1/inputSelector/danteOnRamp "Analog Input"\n

- Response: OK\n
- This will set the Dante On Ramp setting on Channel 1 to "Analog Input"



Amplifier Channel Levels

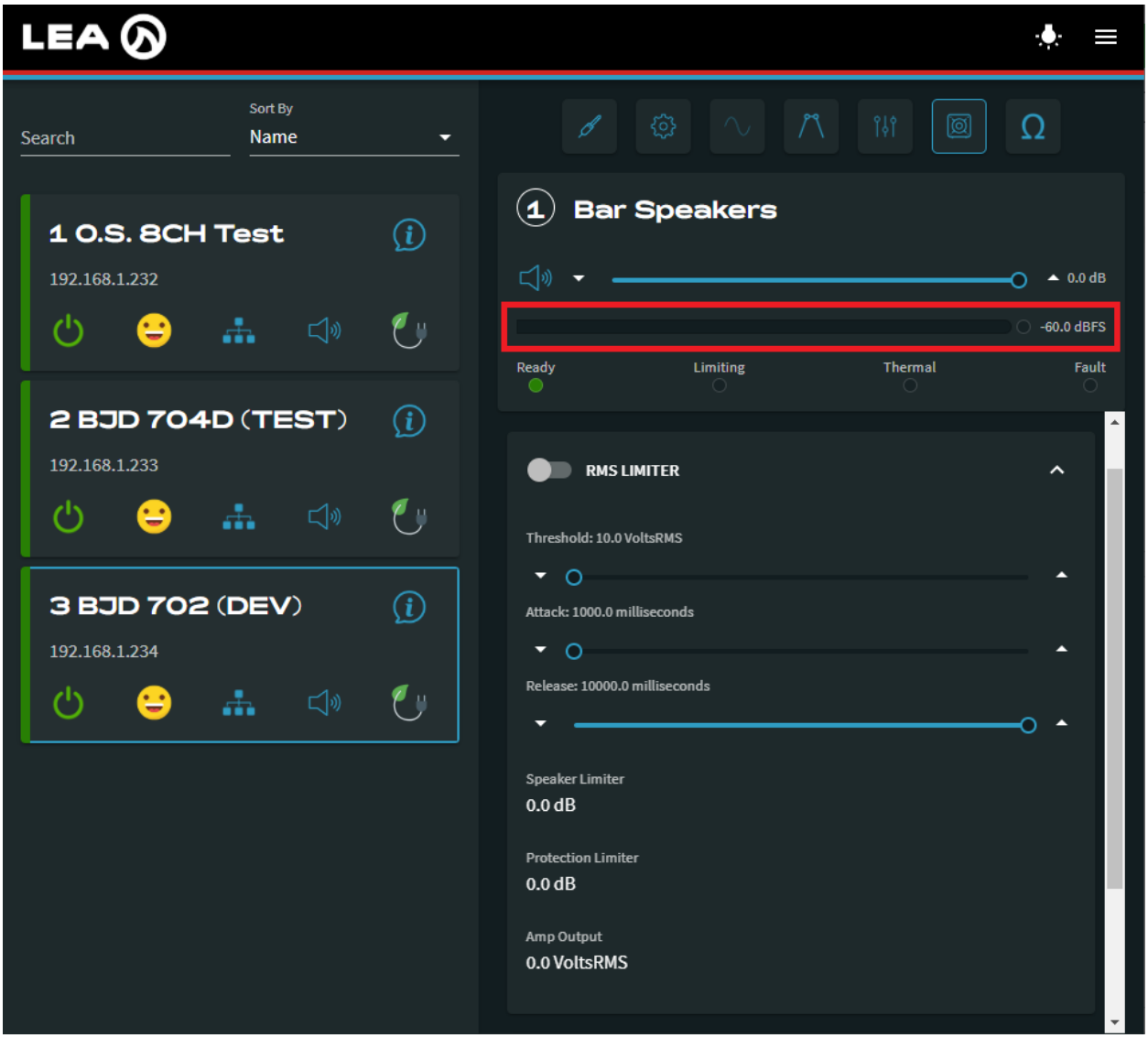
Click on this button to navigate to the Limiter section of the DSP relevant to this API section



Output Channel Meter Level dBFS

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/levels/level_db
Values: -60.0 through 0.0
Example: get /amp/channels/1/levels/level_db\n

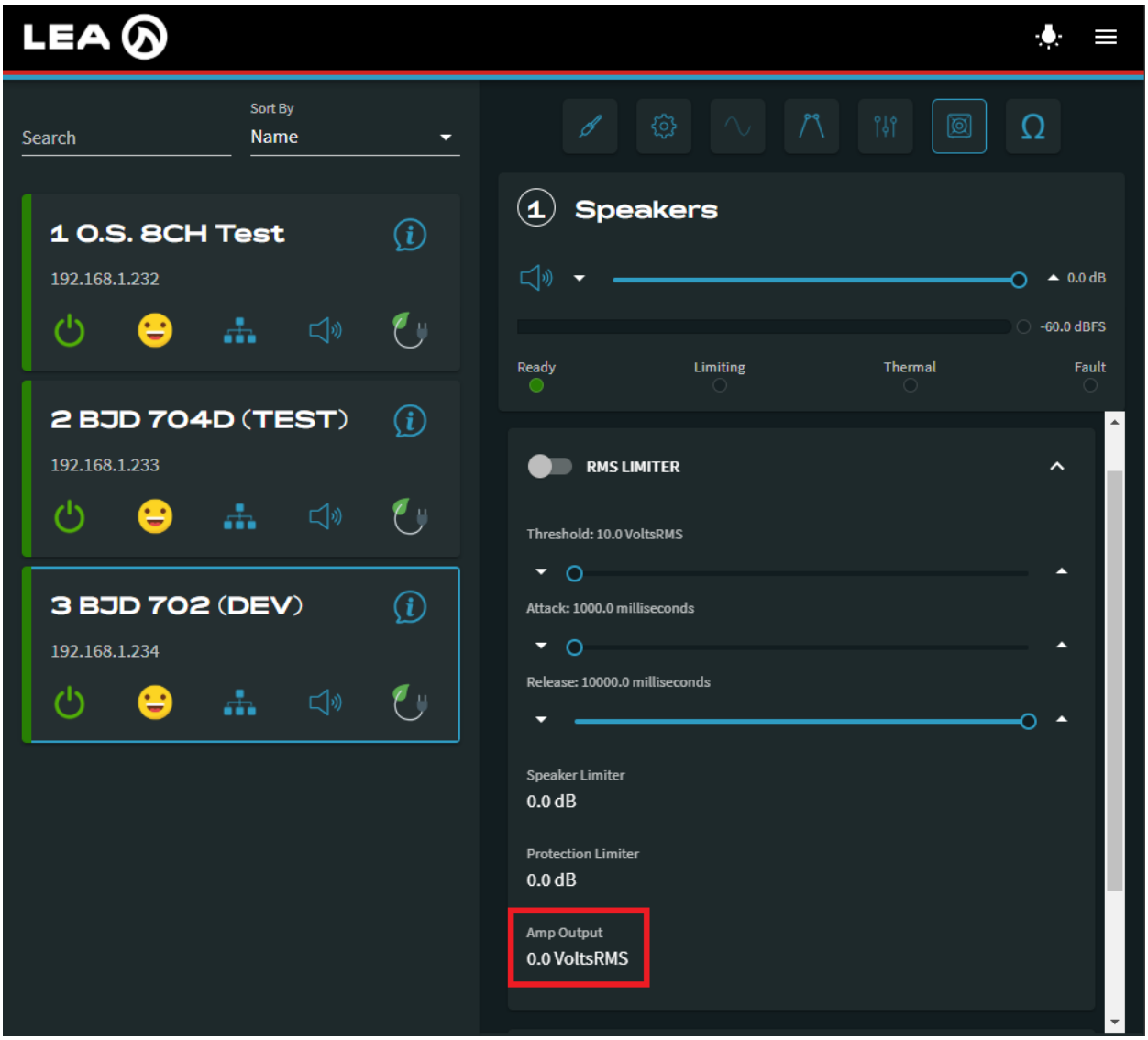
- Response: /amp/channels/1/levels/level_db -31.5\n
    - This command asked for the output level dBFS on channel 1 and got the response -31.5 dBFS



Output Channel Level Volts RMS

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/levels/level_volts
Values: 0.0 through 200.0
Example: get /amp/channels/1/levels/level_volts\n

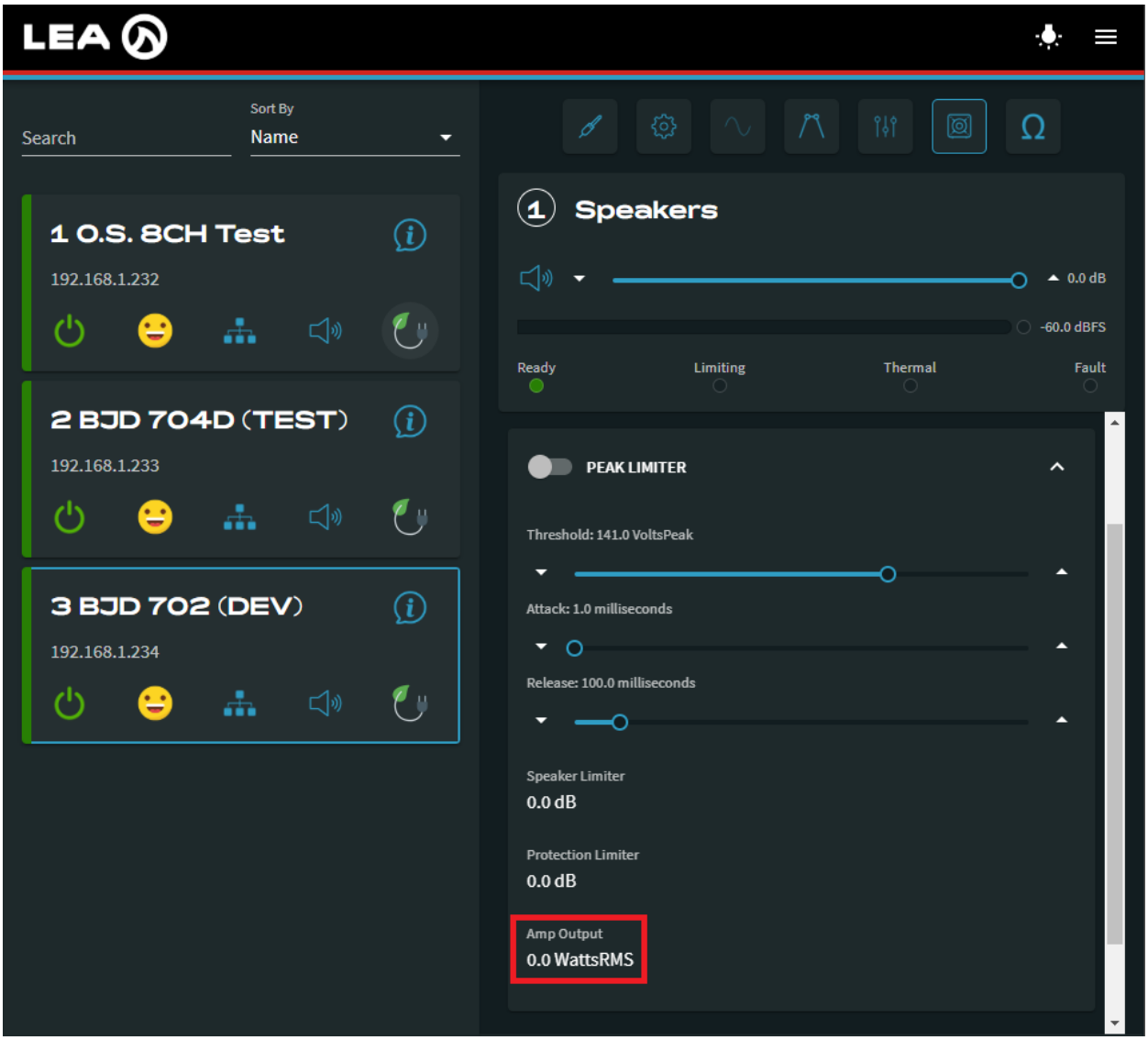
- Response: /amp/channels/1/levels/level_volts 24.5\n
    - This command asked for the output in volts RMS on channel 1 and got the response 24.5 V



Output Channel Level Watts RMS

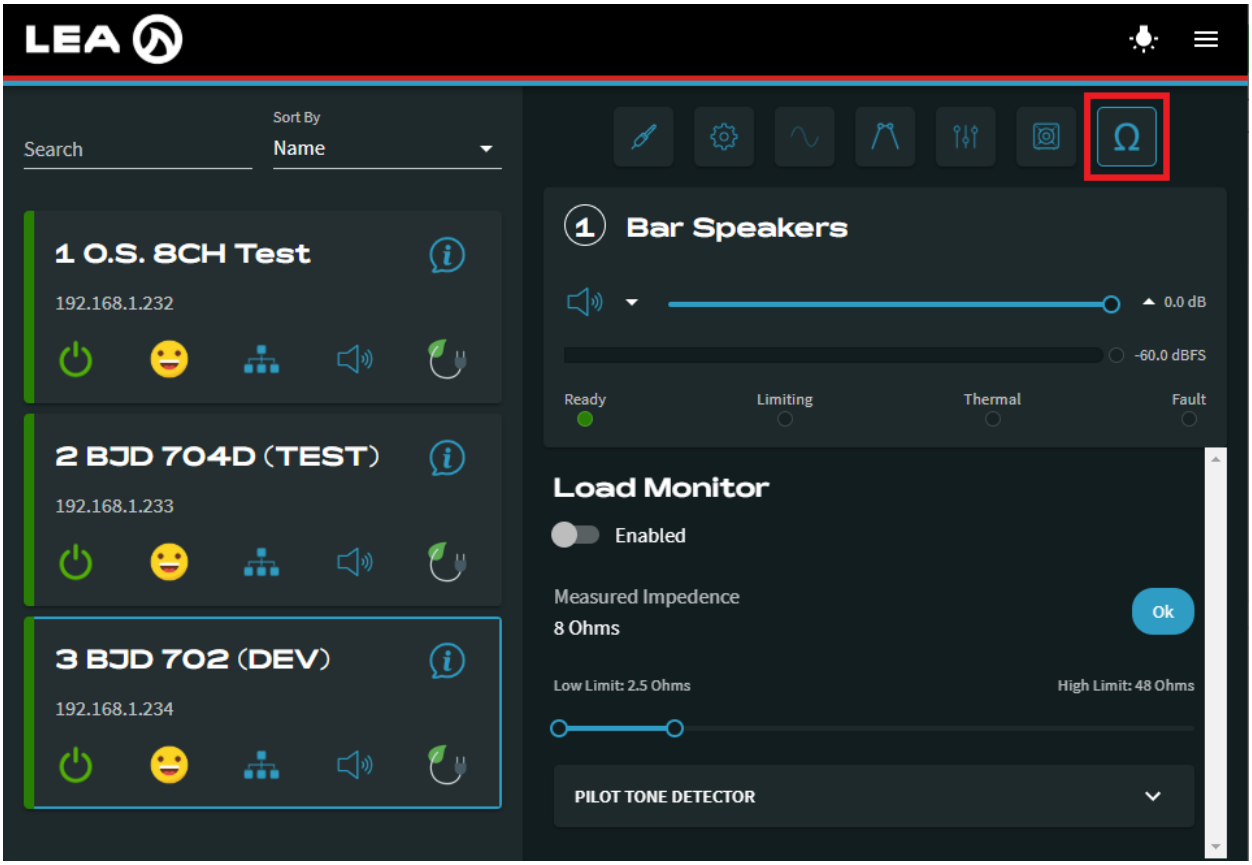
Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/levels/level_watts
Values: 0.0 through 10000.0
Example: get /amp/channels/1/levels/level_watts\n

- Response: /amp/channels/1/levels/level_watts 85.5\n
    - This command asked for the output in watts RMS on channel 1 and got the response 85.5 W



Amplifier Channel Load Monitoring

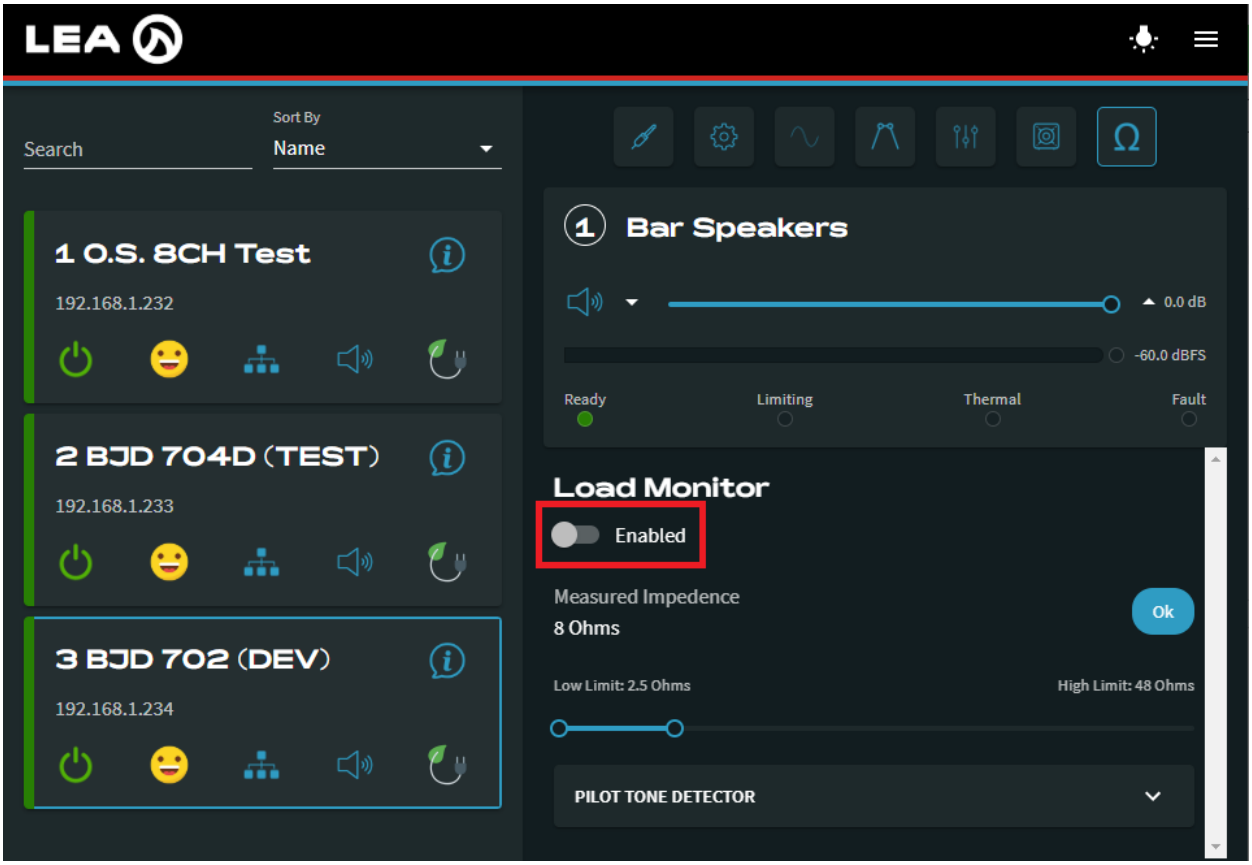
Click on this button to navigate to the Load Monitoring section of the DSP relevant to this API section



Load Monitor Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/loadMonitor/enable
Values: "true", "false"
Example: set /amp/channels/1/loadMonitor/enable "false"\n

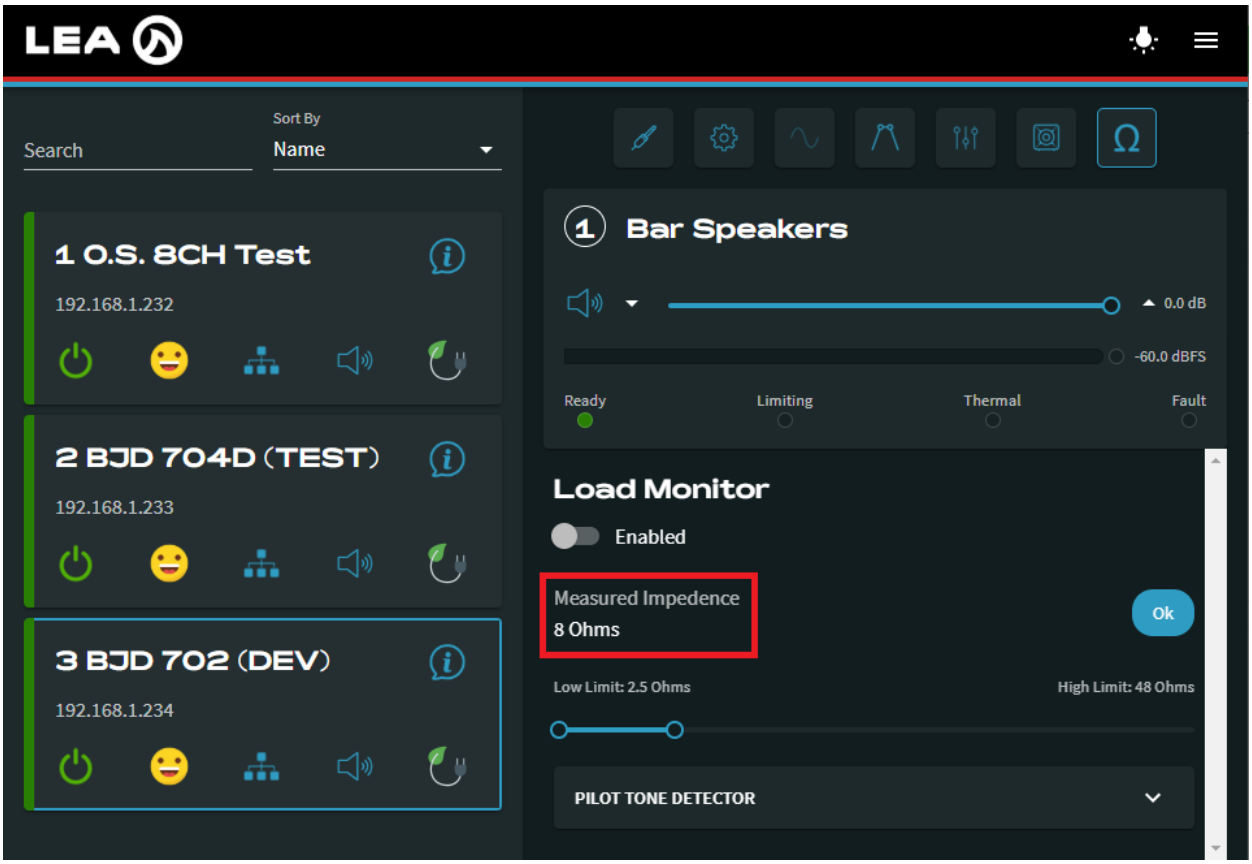
- This command disabled load monitoring on channel 1



Load Monitor Measured Impedance

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/loadMonitor/measuredImpedance
Values: 0.0 through 250.0
Example: get /amp/channels/1/loadMonitor/measuredImpedance\n

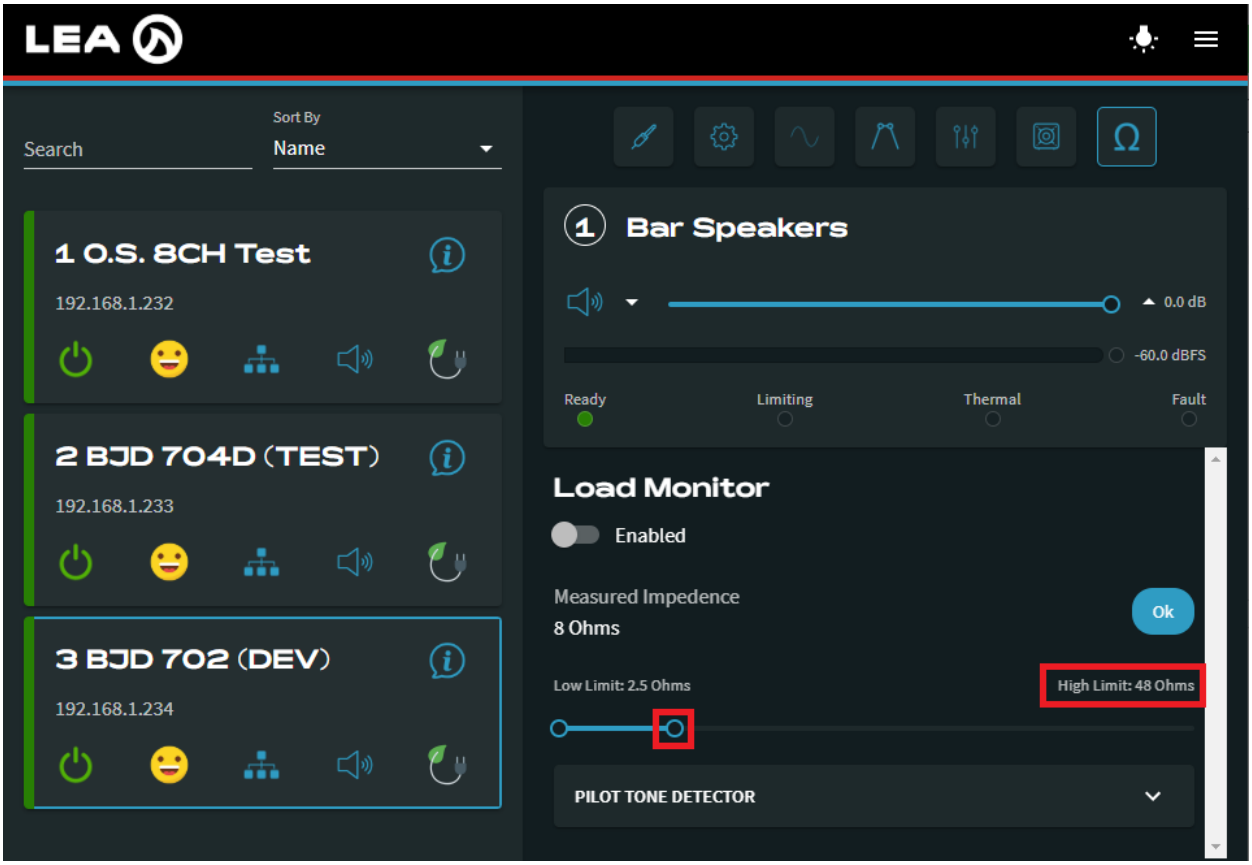
- Response: /amp/channels/1/loadMonitor/measuredImpedance 8.2\n
- This command asked for the impedance measured on channel 1 and got the response 8.2 Ω



Load Monitor High Limit

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/loadMonitor/highLimit
Values: 8.0 through 250.0
Example: set /amp/channels/1/loadMonitor/highLimit 48.0\n

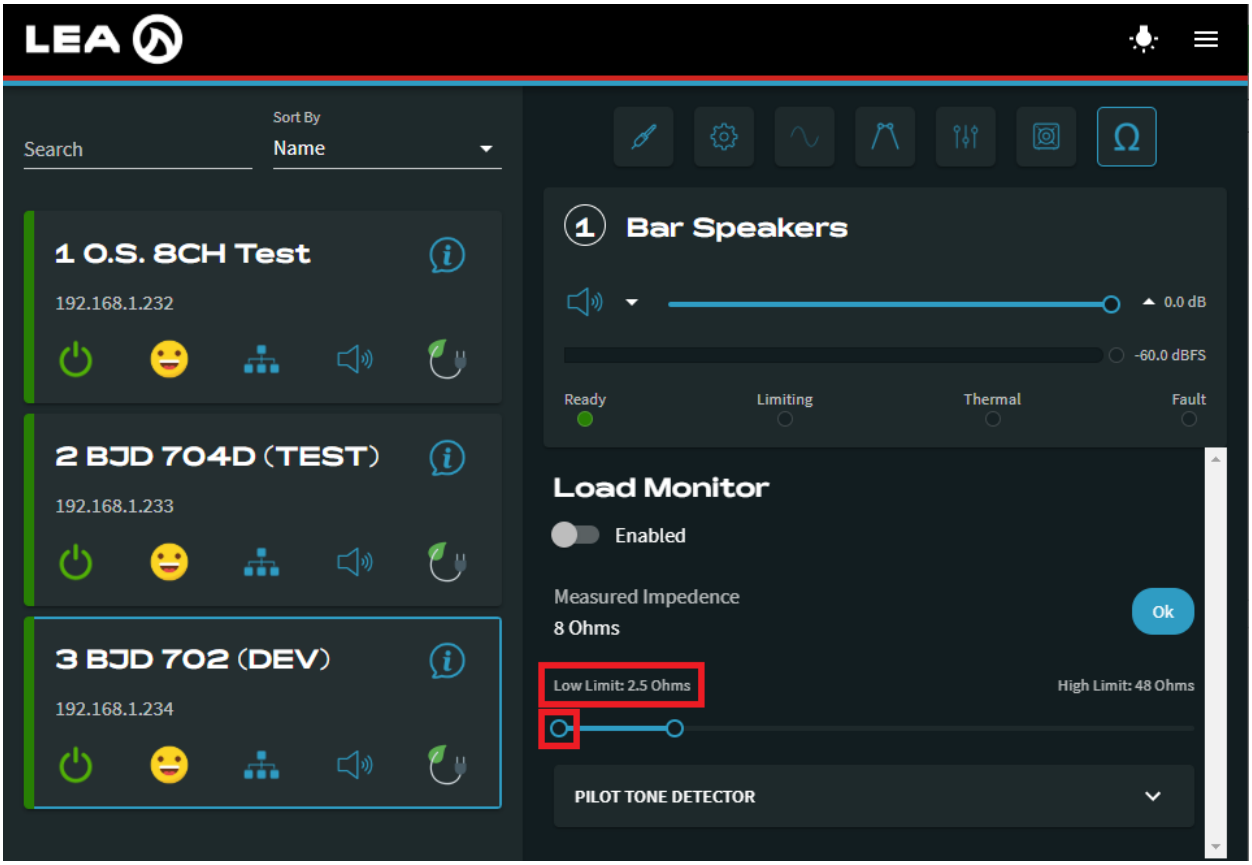
- This command set the load monitor high limit on channel 1 to 48 Ω



Load Monitor Low Limit

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/loadMonitor/lowLimit
Values: 1.0 through 250.0
Example: set /amp/channels/1/loadMonitor/lowLimit 2.5\n

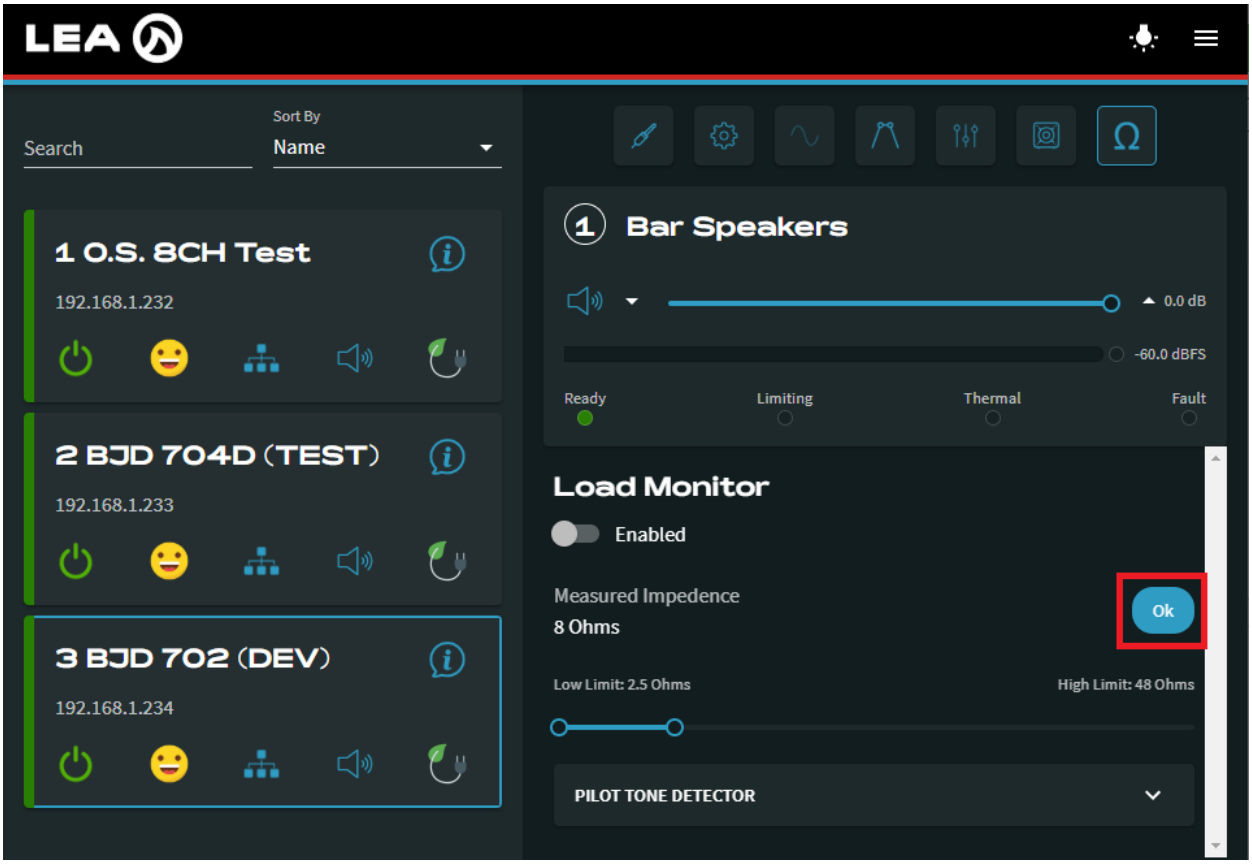
- This command set the load monitor low limit on channel 1 to 2.5 Ω



Load Monitor Status

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/loadMonitor/status
Values: "Ok", "Short", "Open", "Low Signal"
Example: get /amp/channels/1/loadMonitor/status\n

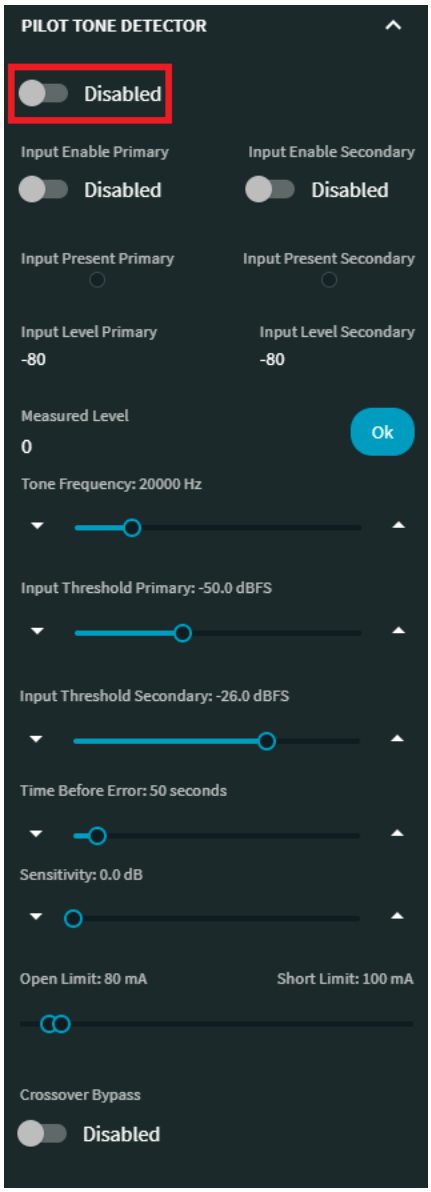
- Response: /amp/channels/1/loadMonitor/status "Ok"\n
- This command asked for the load monitor status and got the response Ok meaning the measured impedance is in between the low and high limits



Pilot Tone Enable

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/enable
Values: "true", "false"
Example: set /amp/channels/1/pilotToneDetector/enable true\n

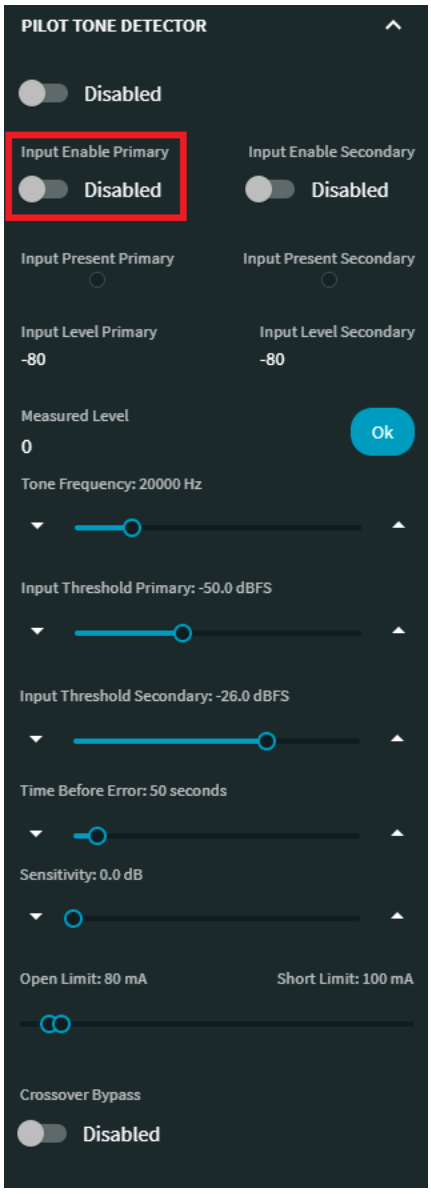
- Response: OK\n
- This command set the Pilot Tone Enable to TRUE or enabled



Pilot Tone Input Enable Primary

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/inputEnablePrimary
Values: "true", "false"
Example: set /amp/channels/1/pilotToneDetector/inputEnablePrimary true\n

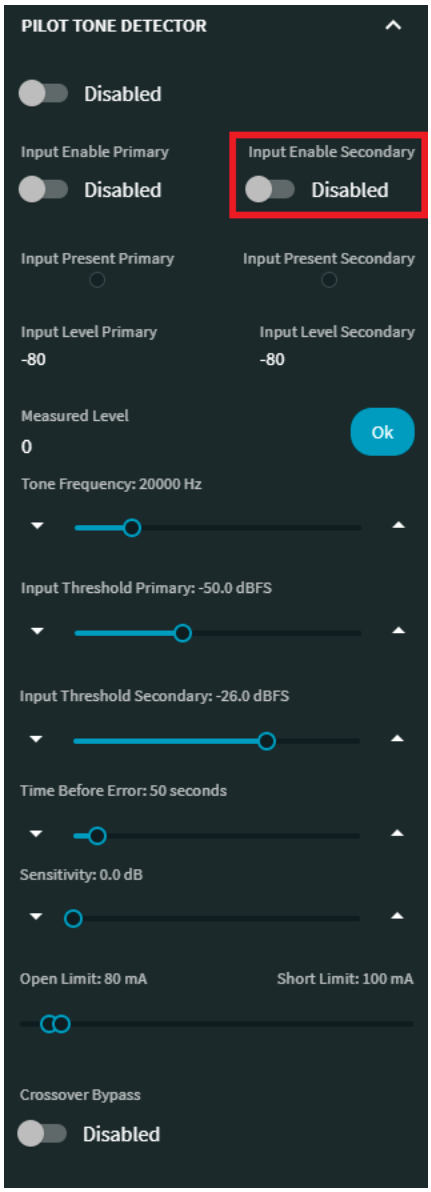
- Response: OK\n
- This command set the Pilot Tone Input Enable Primary to TRUE or enabled



Pilot Tone Input Enable Secondary

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/inputEnableSecondary
Values: "true", "false"
Example: set /amp/channels/1/pilotToneDetector/inputEnableSecondary true\n

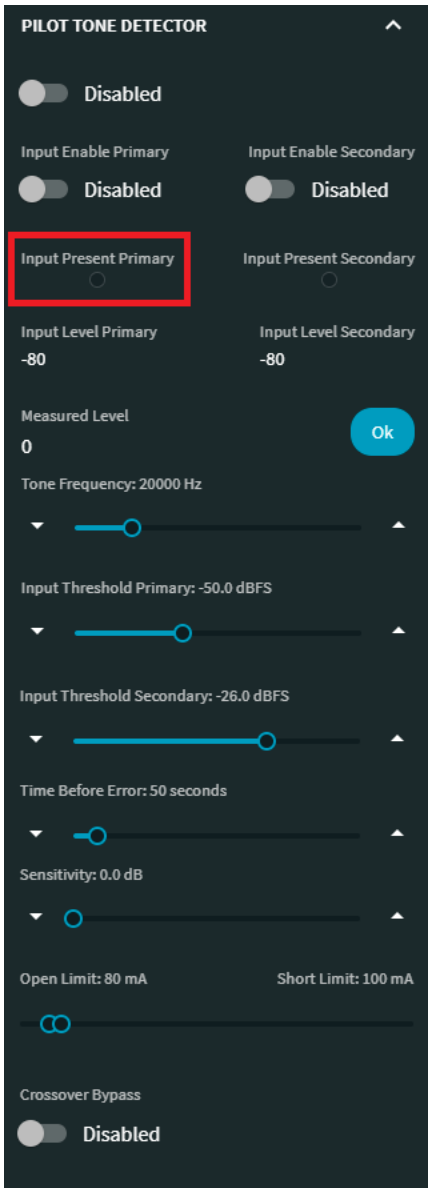
- Response: OK\n
- This command set the Pilot Tone Input Enable Secondary to TRUE or enabled



Pilot Tone Input Present Primary

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/inputPresentPrimary
Values: "true", "false"
Example: get /amp/channels/1/pilotToneDetector/inputPresentPrimary\n

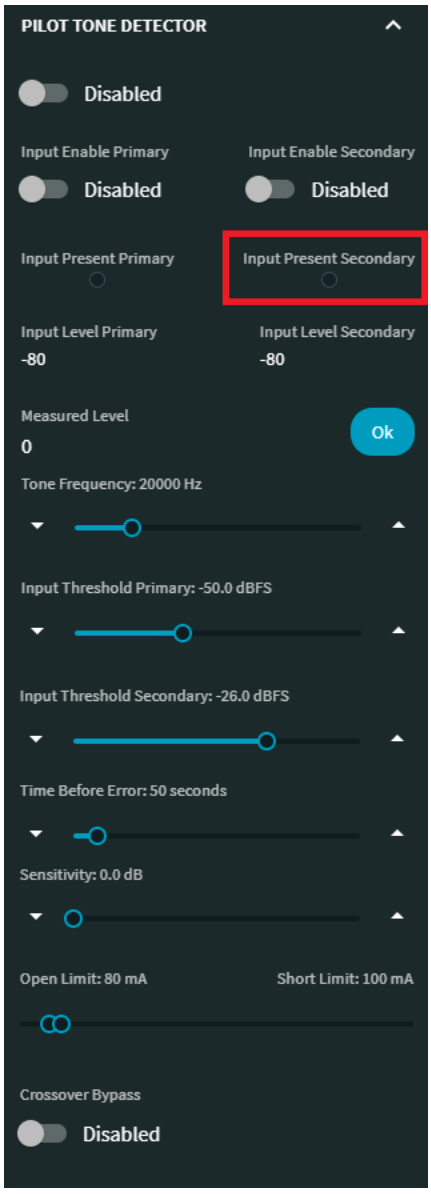
- Response: /amp/channels/1/pilotToneDetector/inputPresentPrimary false\n
- This command asked for the Pilot Tone Input Present Primary state and received FALSE



Pilot Tone Input Present Secondary

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/inputPresentSecondary
Values: "true", "false"
Example: get /amp/channels/1/pilotToneDetector/inputPresentSecondary\n

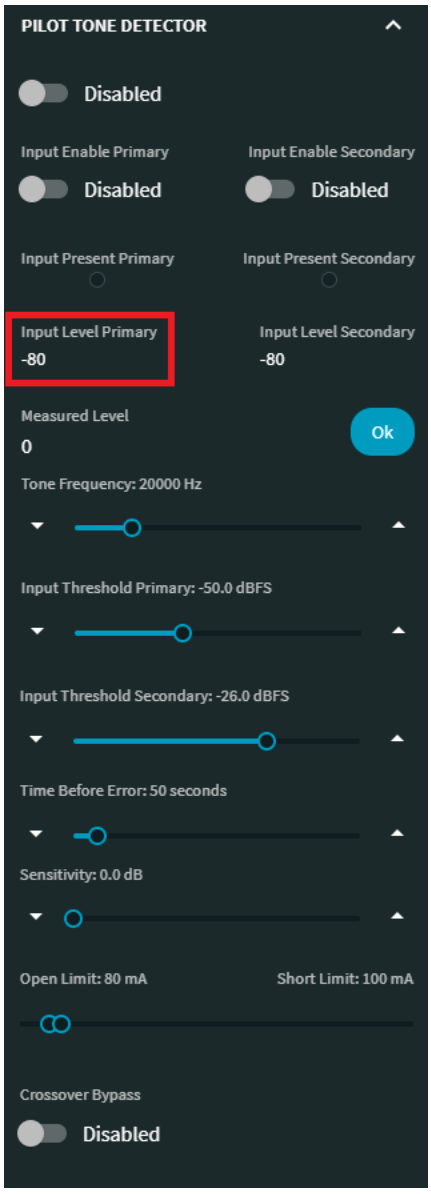
- Response: /amp/channels/1/pilotToneDetector/inputPresentSecondary false\n
- This command asked for the Pilot Tone Input Present Secondary state and received FALSE



Pilot Tone Input Level Primary

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/inputLevelPrimary
Values: -80.0 through 0.0
Example: get /amp/channels/1/pilotToneDetector/inputLevelPrimary\n

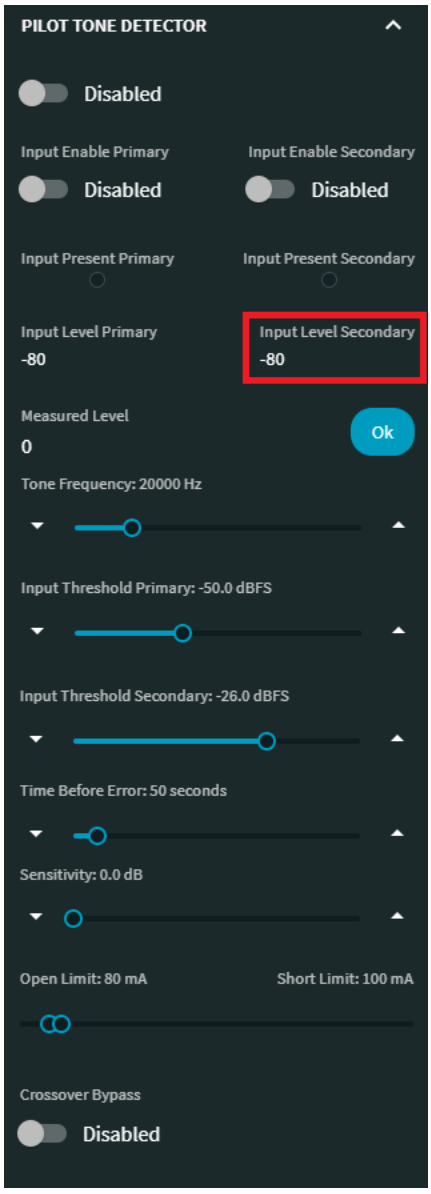
- Response: /amp/channels/1/pilotToneDetector/inputLevelPrimary -33\n
- This command asked for the Pilot Tone Input Level Primary and received -33dB



Pilot Tone Input Level Secondary

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/inputLevelSecondary
Values: -80.0 through 0.0
Example: get /amp/channels/1/pilotToneDetector/inputLevelSecondary\n

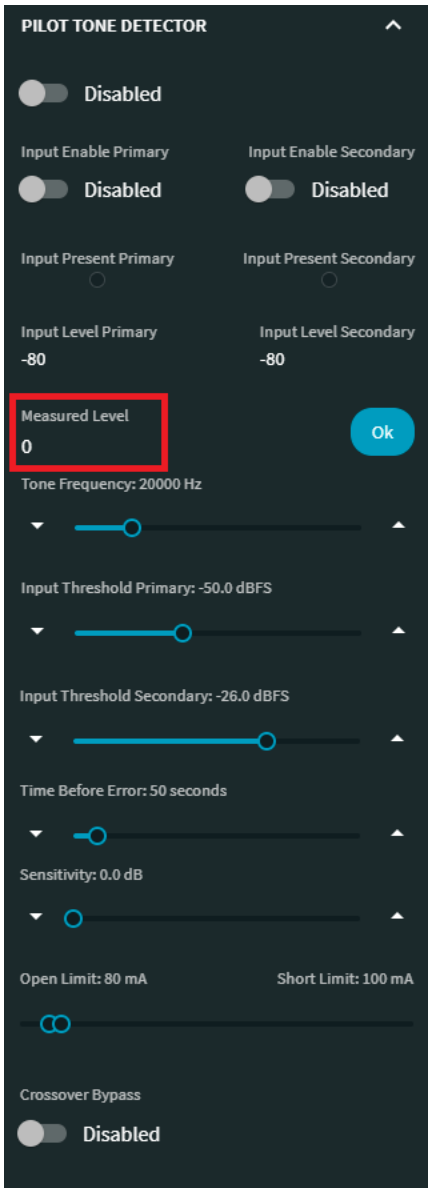
- Response: /amp/channels/1/pilotToneDetector/inputLevelSecondary -38\n
- This command asked for the Pilot Tone Input Level Secondary and received -38dB



Pilot Tone Measured Level

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/measuredLevel
Values: 0 through 65535 mA ptd
Example: get /amp/channels/1/pilotToneDetector/measuredLevel\n

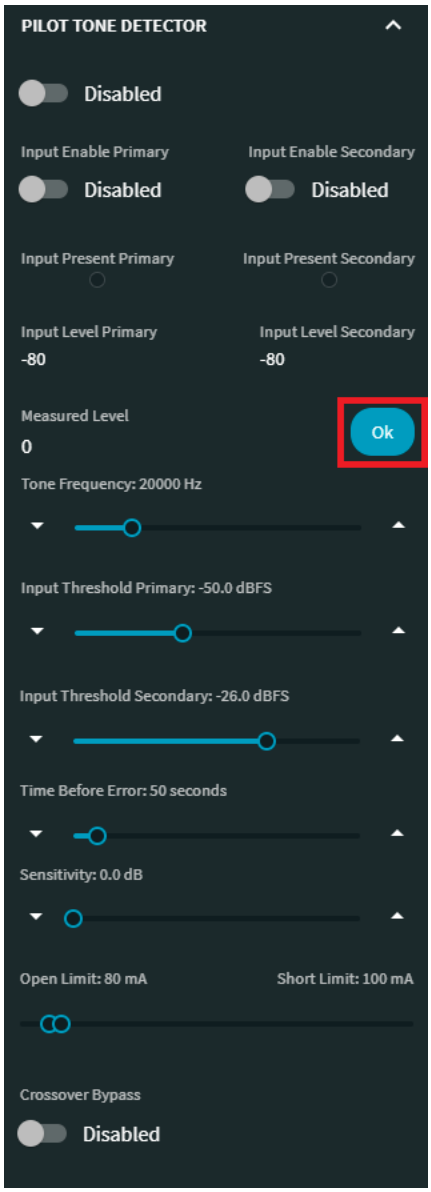
- Response: /amp/channels/1/pilotToneDetector/measuredLevel 35\n
- This command asked for the Pilot Tone Measured Level and received 35mA



Pilot Tone Status

Type: SENSOR
Commands: get, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/status
Values: "Ok", "Short", "Above Short Limit", "Below Open Limit", "Open", "Low Signal"
Example: get /amp/channels/1/pilotToneDetector/status\n

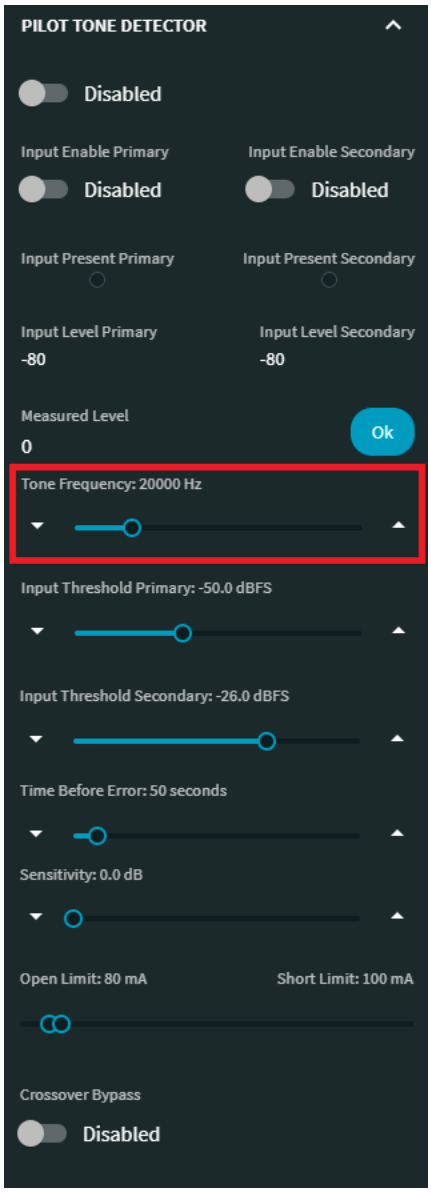
- Response: /amp/channels/1/pilotToneDetector/status "Ok"\n
- This command asked for the Pilot Tone Status and received OK as the response



Pilot Tone Frequency

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/toneFrequency
Values: 19000 through 24000
Example: set /amp/channels/1/pilotToneDetector/toneFrequency 20000\n

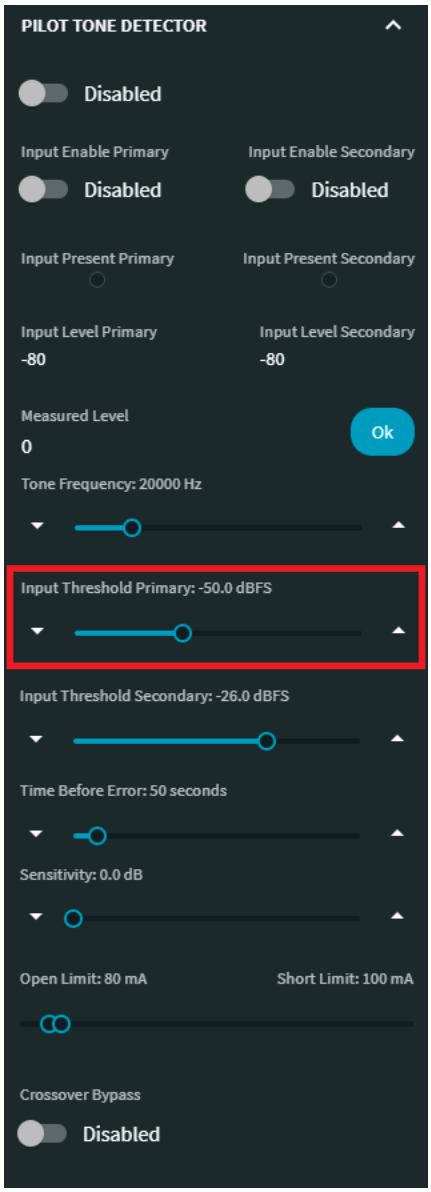
- Response: OK\n
- This command set the Pilot Tone Frequency to 20000Hz



Pilot Tone Input Threshold Primary

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/inputThresholdPrimary
Values: -80.0 through 0.0
Example: set /amp/channels/1/pilotToneDetector/inputThresholdPrimary -50\n

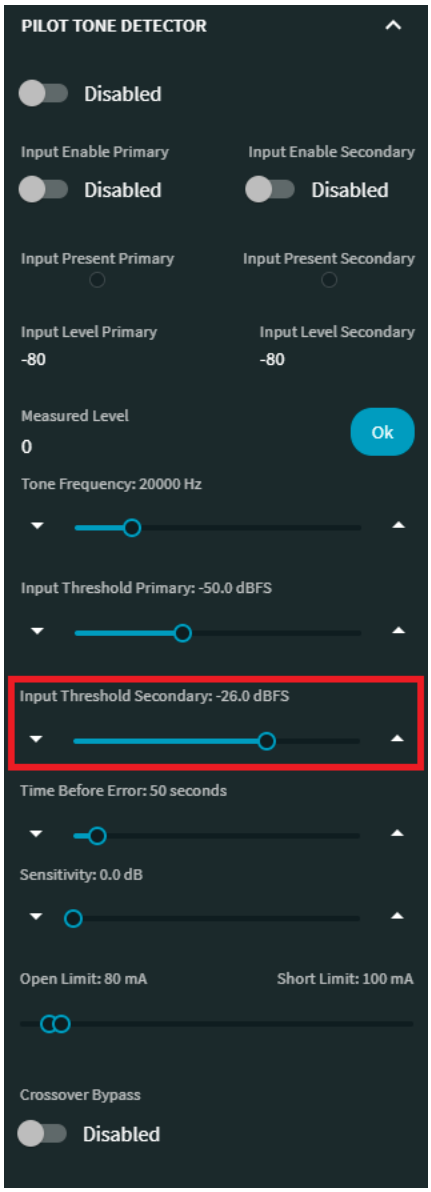
- Response: OK\n
- This command set the Pilot Tone Input Threshold Primary to -50dB



Pilot Tone Input Threshold Secondary

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/inputThresholdSecondary
Values: -80.0 through 0.0
Example: set /amp/channels/1/pilotToneDetector/inputThresholdSecondary -26\n

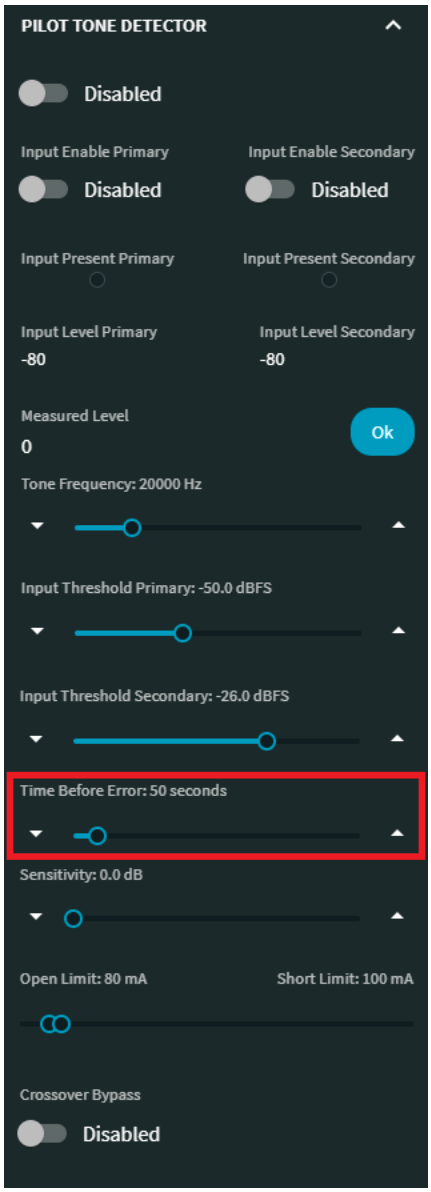
- Response: OK\n
- This command set the Pilot Tone Input Threshold Secondary to -26dB



Pilot Tone Time Before Error

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/ timeElapsedForError
Values: 1 through 600
Example: set /amp/channels/1/pilotToneDetector/ timeElapsedForError 50\n

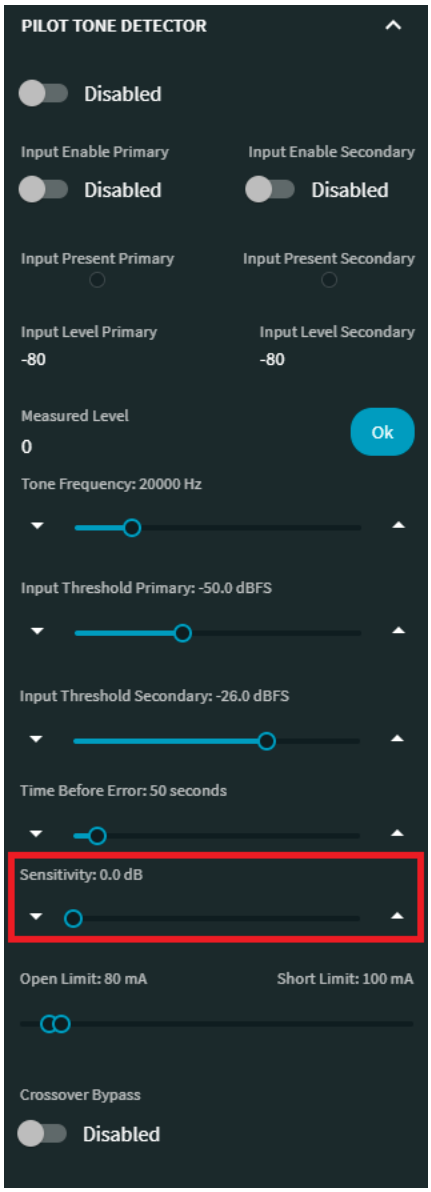
- Response: OK\n
- This command set the Pilot Tone Time Elapsed for Error to 50 seconds



Pilot Tone Sensitivity

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/sensitivity
Values: 0.0 through 12.0
Example: set /amp/channels/1/pilotToneDetector/sensitivity 0\n

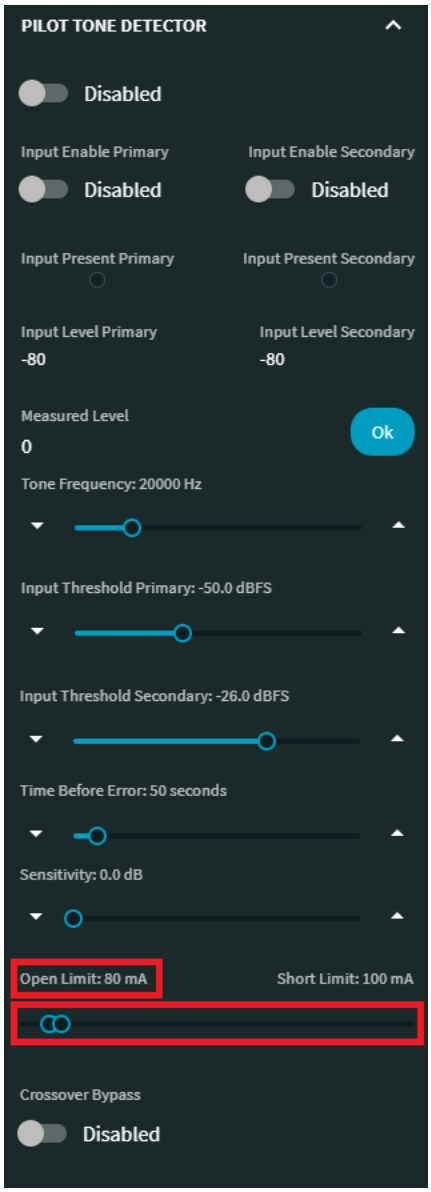
- Response: OK\n
- This command set the Pilot Tone Sensitivity to 0dB



Pilot Tone Open Limit

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/openLimit
Values: 25.0 through 700.0
Example: set /amp/channels/1/pilotToneDetector/openLimit 80\n

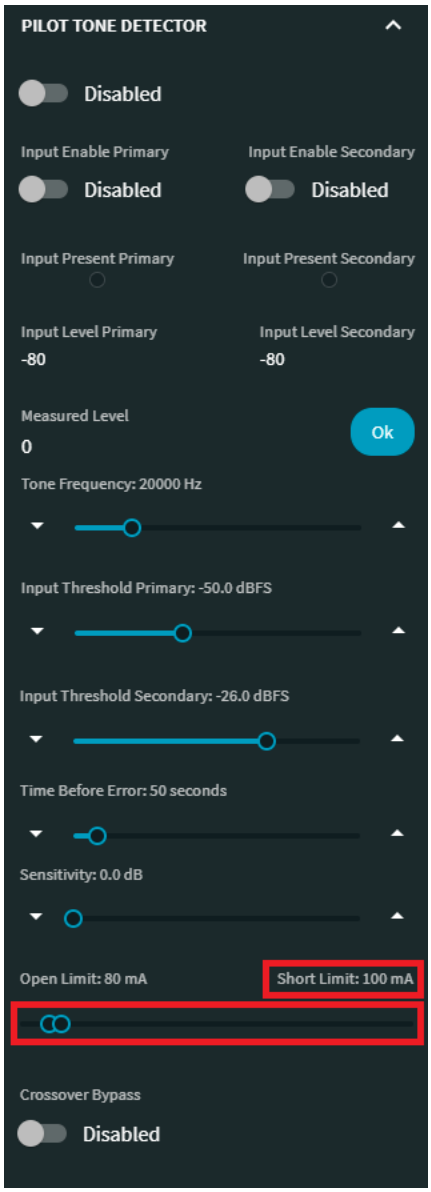
- Response: OK\n
- This command set the Pilot Tone Open Limit to 80mA



Pilot Tone Short Limit

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/shortLimit
Values: 50.0 through 750.0
Example: set /amp/channels/1/pilotToneDetector/shortLimit 100\n

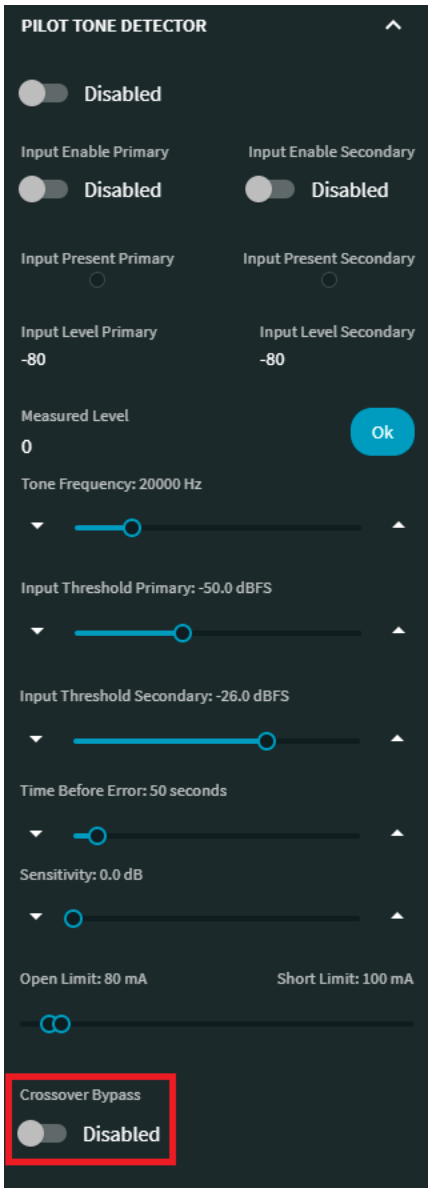
- Response: OK\n
- This command set the Pilot Tone Short Limit to 100mA



Pilot Tone Crossover Bypass

Type: CONTROL
Commands: get, set, subscribe, unsubscribe
URL: /amp/channels/x/pilotToneDetector/CrossoverBypass
Values: "true", "false"
Example: set /amp/channels/1/pilotToneDetector/CrossoverBypass false\n

- Response: OK\n
- This command set the Pilot Tone Crossover Bypass to FALSE or DISABLED



Revision History

Rev	Date	Changed by	Description
0	4-17-2020	DB &WAP	Initial
1	5-11-2020	BJD	Addition of Available Commands
2	10-02-2020	BJD	Corrected some typos and added additional calls to Device Info
3	10-15-2020	BJD	Corrected some example responses that were incorrect
4	11-02-2020	BJD	Changed an example command to a more common example
5	09-21-2021	BJD	Updated for 2.0.2.X Firmware: Dante On Ramp, Pilot Tone, Priority Override Threshold
5.1	12-16-2021	BJD	Typo Correction
6	05-19-2022	NM	Corrected example responses that were incorrect