



You make **possible**



# Webex Room Device APIs: The power of automated, integrated, immersive collaboration

Richard Bayes – Product Manager  
@CollabRich

DEVNET-1462

**CISCO** *Live!*

Barcelona | January 27-31, 2020



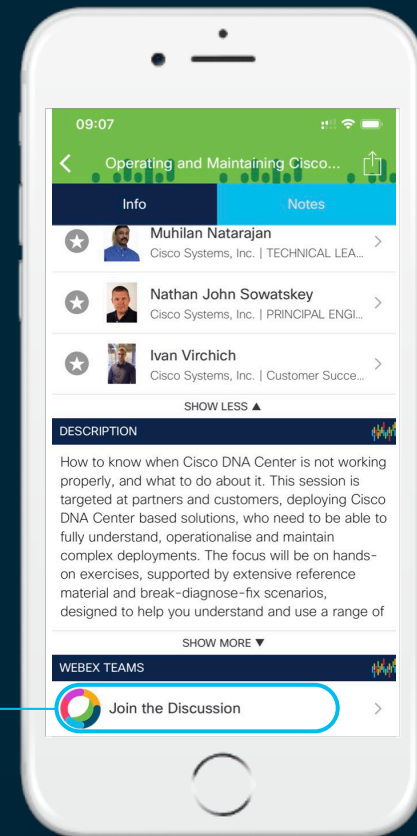
# Cisco Webex Teams

## Questions?

Use Cisco Webex Teams to chat with the speaker after the session

## How

- 1 Find this session in the Cisco Events Mobile App
- 2 Click “Join the Discussion”
- 3 Install Webex Teams or go directly to the team space
- 4 Enter messages/questions in the team space



# Agenda

- Introduction
- Top Level API Overview
- Transport Methods and API Types
- New Features Released in the last 12 Months
- Business Use Case Examples
- Roadmap & Future State
- Conclusion



<https://projectworkplace.cisco.com>

# What are APIs?

We hear about how valuable APIs are all the time, people often say that they have a significant impact on business. However, what is the definition of an API?

- Application Programable Interface
- A defined a way to access features and data from services
- There are over 21,000 Public APIs attached to services in the world today!

# The Open Platform

The image shows a bright, modern office space with large windows in the background. In the foreground, there is a long wooden table with several wooden chairs. A person is sitting at the table, facing away from the camera. In the background, other people are visible, some standing and some sitting. The entire image is overlaid with a blue tint. The text "The Open Platform" is written in white, centered over the image.

Enabling customers to create customizations that attach to their existing workflows.

Allowing partners to create additional value beyond the Webex core offering.



# Open Platform = Open APIs

- Supply easy to consume documentation for developers and partners – Work has started on a searchable API webpage
- Creating APIs for Webex Control Hub to bring features to cloud deployments
- Promoting usecases that solve business problems
- Develop new features with an API first mindset.

# Ways to Interact with xAPI on the Device Level

- Terminal Mode (SSH)
- XML
- HTTP/HTTPS
- JSXAPI (New Release 5.0!)
- Macros
- Websockets (NEW 2019!)
- Cloud xAPI / REST (NEW 2019!)
- Commands (xCommand)
- Configurations (xConfiguration)
- Statuses (xStatus)
- Events (xFeedback)

```
OK
xconfiguration ?
- User Configurations -

Audio          Logging          Proximity        UserInterface
CallHistory     Macros           RoomAnalytics    UserManagement
Cameras         Network          SerialPort       Video
Conference      NetworkServices SIP              WebEngine
GPIO            Peripherals     Standby
H323            Phonebook       SystemUnit
HttpClient      Provisioning     Time

OK
xstatus ?
- Status -

Audio          GPIO            Peripherals      Standby
Bookings       H320            Provisioning     SystemUnit
Call           H323            Proximity        Time
Cameras        HttpFeedback    RoomAnalytics    UserInterface
Capabilities    MediaChannels   RoomPreset       Video
Conference      Network         Security
Diagnostics     NetworkServices SIP

OK
xcommand ?
- User Commands -

Audio          Diagnostics      Peripherals      Standby
Bookings       Dial             Phonebook        SystemUnit
Call           GPIO             Presentation      Time
CallHistory    HttpClient       Provisioning     UserInterface
Camera         HttpFeedback     Proximity        UserManagement
Cameras        Macros           RoomPreset       Video
Conference     Message          Security
```

OK

## HTTP API

### putxml

http://IPADDRESS/putxml

```
<Configuration>  
  <RoomAnalytics>  
    <PeoplePresenceDetector>On</PeoplePresenceDetector>  
  </RoomAnalytics>  
</Configuration>
```

## HTTP API

### getxml

`http://IPADDRESS/getxml?location=/Status/RoomAnalytics/PeopleCount`

```
<?xml version="1.0"?>
<Status product="Cisco Codec" version="ce9.8.0" apiVersion="4">
  <RoomAnalytics>
    <PeopleCount>
      <Current>3</Current>
    </PeopleCount>
  </RoomAnalytics>
</Status>
```

# Macro editor

Import from file...

Create new macro

In-Room Control Debugger ☒

Speed Dials ☒

Room Reset ☒

Microphone Controls ☒

Untitled1 ☐


Untitled2 ☐

Apple TV Controls ☐

SX80 GPIO ☒

Room Setup ☐

Customer Satisfaction Measure ☐



Runtime

Preferences

Help

```
1 const xapi = require('xapi');
2 var democounter = 0;
3
4
5 xapi.status.on('SystemUnit State NumberOfActiveCalls', (callCounter) => {
6   if(callCounter == 1){
7     xapi.command("UserInterface Message Alert Display", {Text: 'OK, you are all set. Have a great meeting', Title: 'Meeting Started Confirmation', Duration:3});
8   }
9 });
10
11
12 xapi.event.on('UserInterface Extensions Widget Action', (event) => {
13   if(event.WidgetId == 'meetingpreset_david'){
14     if(event.Type == 'clicked'){
15       console.log("David Bruun-Lie");
16       xapi.command("dial", {number: 'com'});
17       democounter = 1;
18     }
19   }
20   if(event.WidgetId == 'meetingpreset_demoroom'){
21     if(event.Type == 'clicked'){
22       console.log("Dial experience Review demo");
23       xapi.command("dial", {number: 'kafka_hopen_38com'});
24       democounter = 0;
25     }
26   }
27 });
28
29
30
```

Severity | Macros | ☐ Show history


v1.0.2


# Node.js

CISCO *Live!*

Nutella Per Minute

npm Enterprisefeaturespricingdocumentationsupport



sign up or log in

It's your turn. Help us improve JavaScript. [Take the 2017 JavaScript Ecosystem Survey »](#)

★ jsxapi public

A set of tools to integrate with the Cisco Telepresence Endpoint APIs in JavaScript.

### Documentation

The full API documentation can be built by running `npm install` in a `jsxapi` module directory. Documentation will be located under `docs/` can then be opened in a browser.

More specifically:

```
mkdir tmp
cd tmp
npm install jsxapi
cd node_modules/jsxapi
npm install
```

Then open `./docs/index.html`.

### Example with TSH over SSH

```
const jsxapi = require('jsxapi');

// Connect over ssh to a codec
const xapi = jsxapi.connect('ssh://host.example.com', {
  username: 'admin',
  password: 'password',
});

// Set up a call
xapi.command('Dial', { Number: 'user@example.com' });

// Fetch volume and print it
xapi.status
  .get('Audio Volume')
  .then((volume) => { console.log(volume); });

// Set a configuration
```

 npm install jsxapi

[how? learn more](#)

 myme published 5 days ago

4.0.1 is the latest of 2 releases

[github.com/cisco-ce/jsxapi](https://github.com/cisco-ce/jsxapi)

MIT

Collaborators [list](#)



### Stats

8 downloads in the last day

61 downloads in the last week

61 downloads in the last month

3 open issues on GitHub

No open pull requests on GitHub

### Try it out

 Test jsxapi in your browser.

### Keywords

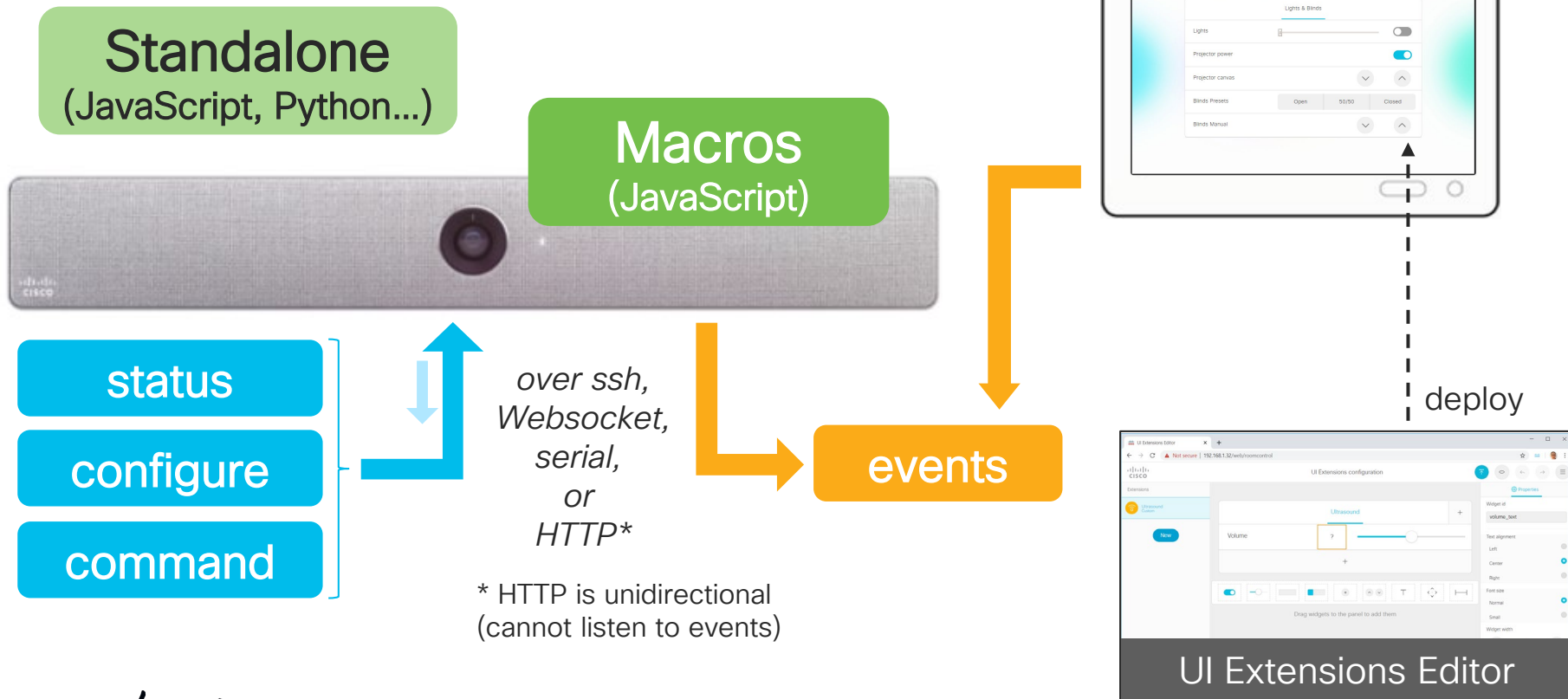
None

### Dependencies (9)

[commander](#), [duplex-passthrough](#), [duplexer](#), [jsonparse](#), [loglevel](#), [ssh2](#), [url-parse](#), [ws](#), [xml-escape](#)

[Renew](#) is hiring. [View more...](#)

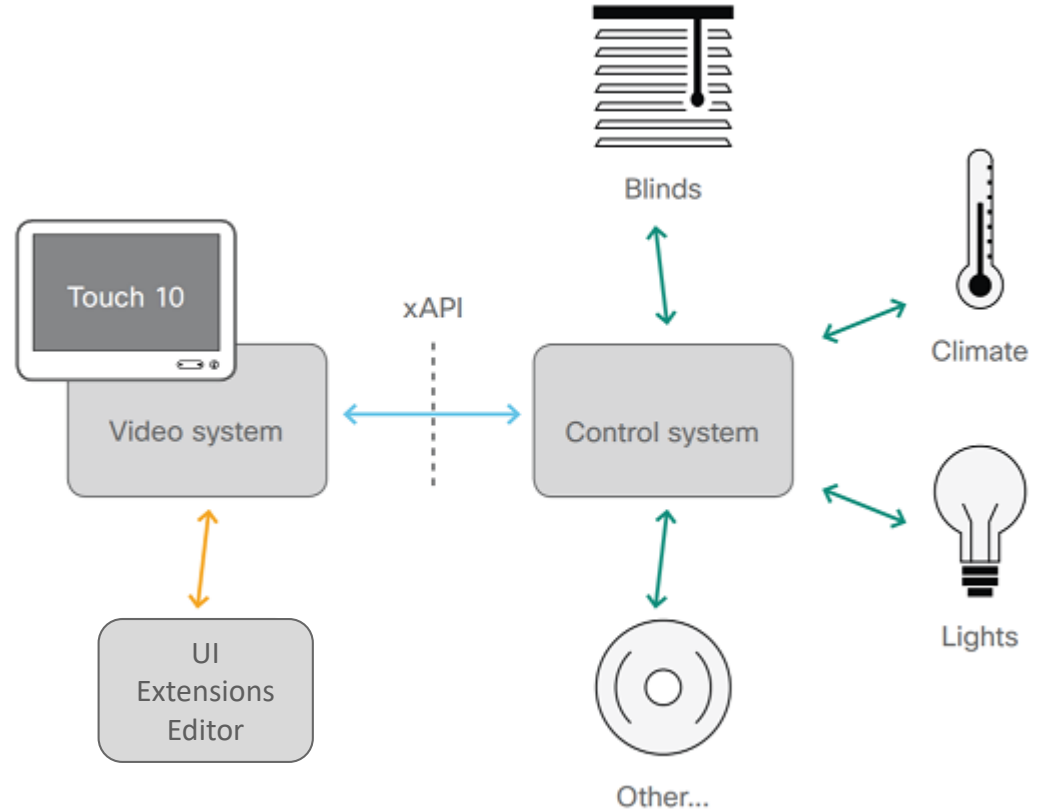
# CE Programmability (xAPI)





# CE Programmability at a glance

- Enhance the User eXperience of your meetings
  - add UI Extensions to Room/Desk/Boards devices
  - interact with IoT devices
  - customized behaviors as macros
  - Integrate with enterprise process
- Automate your devices from code
  - initiate calls, fetch history
  - collect room analytics
  - apply branding across devices
  - deploy UI Extensions & macros



# Cisco Collaboration Devices Programmability

Available on all devices running CE & RoomOS



Room Kit standard, pro, mini

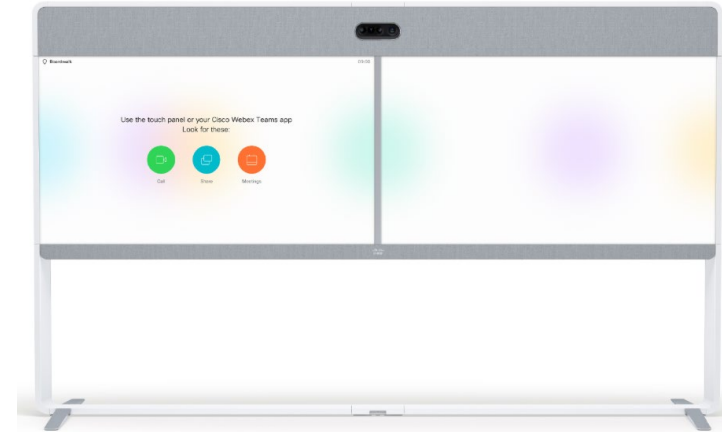


\*no Macros on SX10



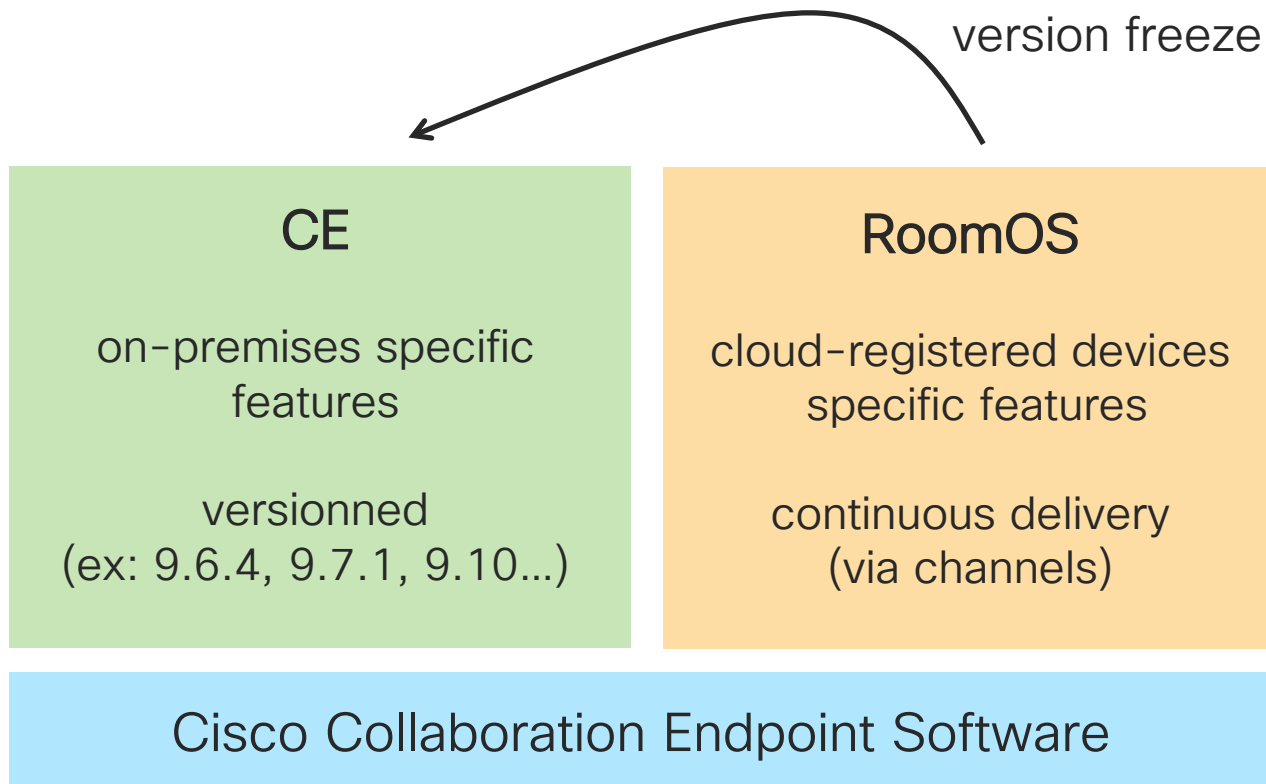
Complete programmability

DX, DeskPro

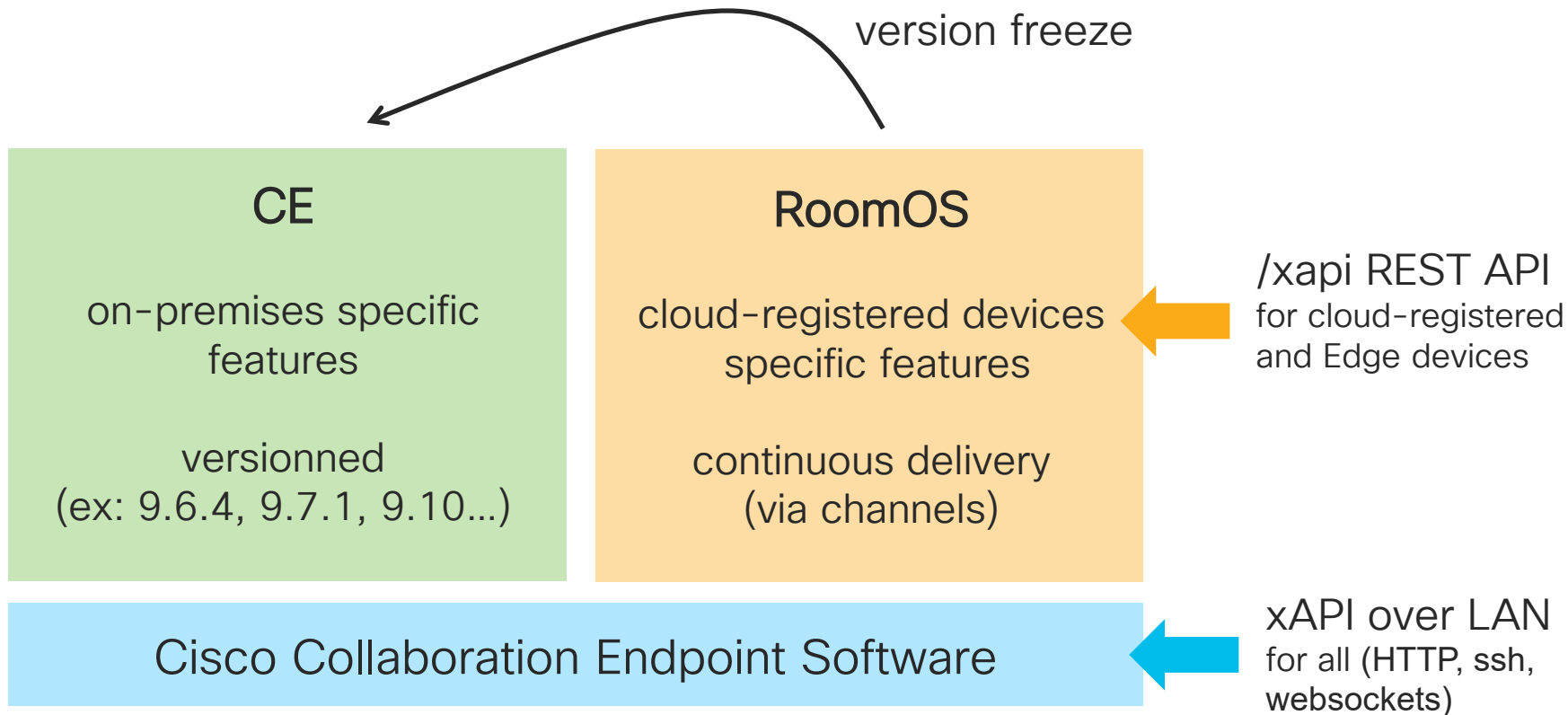


**cisco** *Live!*

# CE and RoomOS

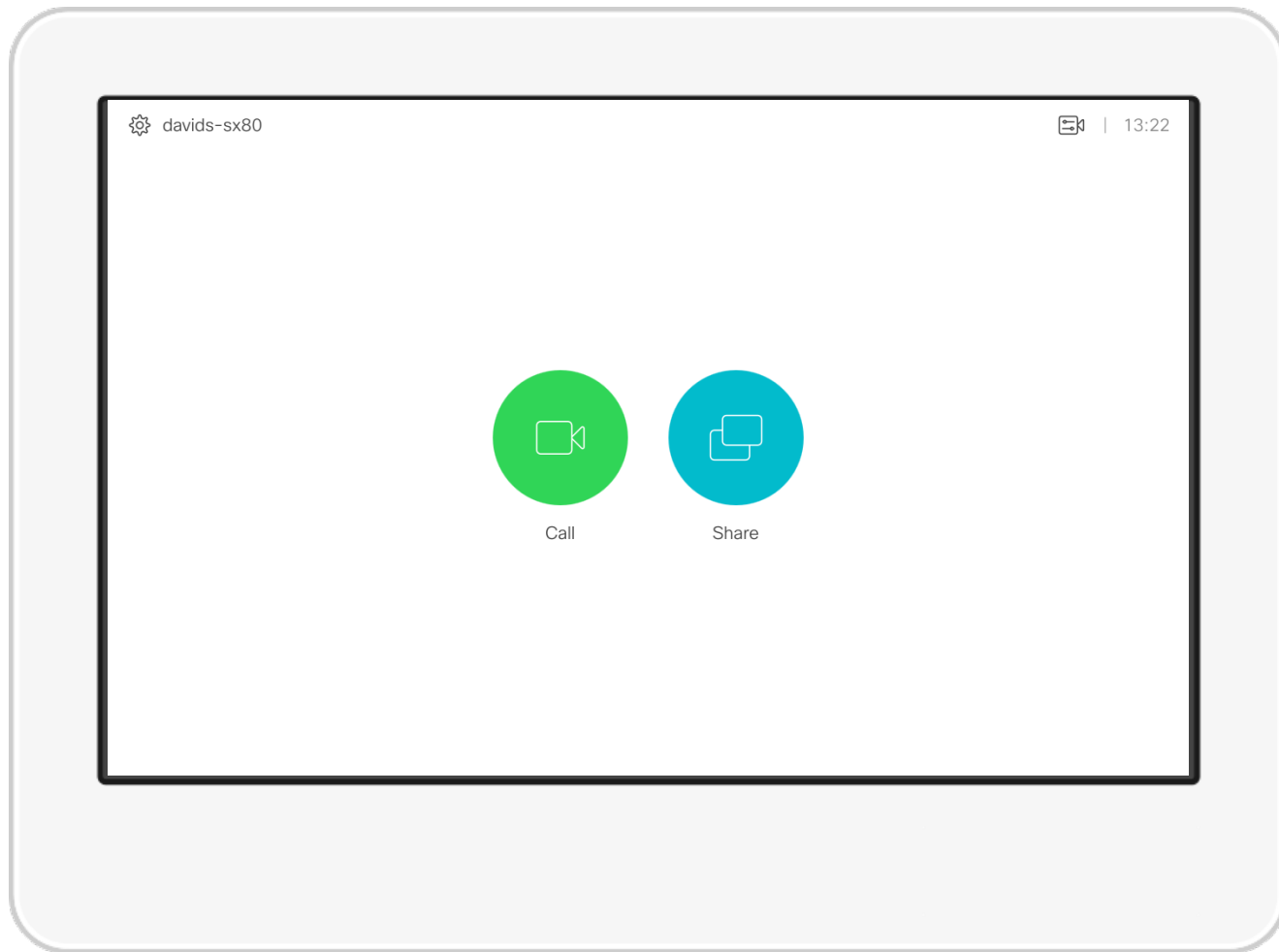


# CE and RoomOS



# Combing the xAPI with In Room Controls





davids-sx80 >

 | 14:51



Call



Share



Lights and Blinds

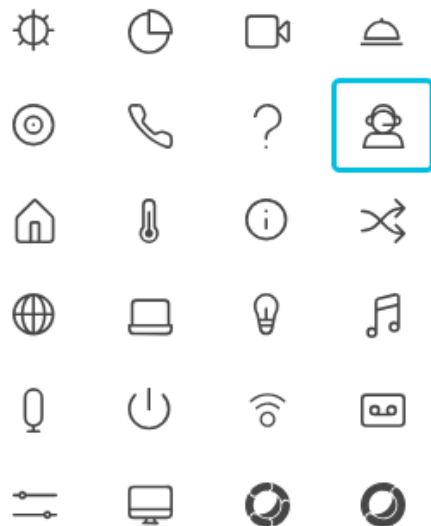


Apple TV



# Icons and Colors

Icon



Panel button color



LYS01-4-RUBIK >

 | 09:25



Call



Share



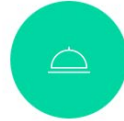
Meetings



Language++



Lights and Blinds



Timer



Camera Layouts

LYS01-4-RUBIK >

 | 09:22

Lights

Blinds

About

Square Light



Downlight



Lights

Blinds

About

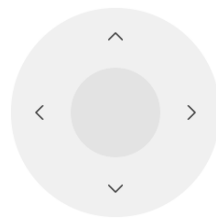
### Blinds Controls



This controls the blinds on the outside of the building. Great for those sunny days!

There may be a slight delay from when you press the controls to the blinds start moving.

### Apple TV Controls



davids-sx80 >

🗨️ | 12:44

How was your recent meeting  
experience?

Fantastic

Good

Not Great

What was new in  
2019!

# New API Additions

- Ability to use 3rd Party HID Devices (Keyboards)
- Video Mute APIs
- Ability to Interact with Webex Cloud Meeting Join Flow
- HTTP POST / GET / PATCH / DELETE
- Additional Video Compositing Options
- Room Analytics (Noise Levels)
- Hide Default Cisco Buttons (Call , Share, Settings)



# Web Engine – Power of Web Applications

- Support for Digital Signage – ability to display any HTML Webpage on the device instead of halfwake.
- Support for Web Applications with xAPI commands
- Interactivity Support for the Webex Board and Desk Pro.



# Cloud / Hybrid Device Management APIs

- Cloud xAPI – Ability to send commands and gather data from devices over a secure cloud connection
- API to allow the creation of Activation Codes
- API to Migrate On Premise Devices to Cloud while preserving key settings
- Key device data visible in Control Hub also being accessible via APIs

[Documentation](#) [Blog](#) [Support](#)

Try it

Example

Request

Response

```
{
  "deviceId": "Y21zY29zcGFyazovL3VzL0RFVklDR",
  "result": {
    "Audio": {
      "Volume": 75
    }
  }
}
```

### Query Status

Query the current status of the Webex RoomOS Device.  
Specify the target device in the `deviceId` parameter in the URI.  
The target device is queried for statuses according to the expression in the `name` parameter.  
See the [xAPI Guide](#) for a description of status expressions.

**GET** `/v1/xapi/status`

Query Parameters

Name	Description
<code>deviceId</code> <small>string Required</small>	The unique identifier for the Webex RoomOS Device.
<code>name</code> <small>string Required</small>	Status expression used to query the Webex RoomOS Device.

Response Properties

Name	Description
<code>deviceId</code> <small>string</small>	The unique identifier for the Webex RoomOS Device.
<code>result</code> <small>object</small>	xAPI status result

Response Codes

The list below describes the common success and error responses you should expect from the API.

# /Devices/ API – <https://developer.webex.com>

```
1  {"id": "Y2lzY29zcGFya290eSA1",
2   "displayName": "LYS01-4-TheAPICave (RB)",
3   "placeId": "Y2lzY29zcGFya290eSA1",
4   "orgId": "Y2lzY29zcGFya290eSA1",
5   "capabilities": [],
6   "permissions": [
7     "xapi"
8   ],
9   "product": "Cisco Webex Board 70",
10  "tags": [],
11  "ip": "10.10.10.10",
12  "mac": "70:1F:53:33:33:33",
13  "serial": "123456789",
14  "activeInterface": "LAN",
15  "software": "RoomOS 2019-05-11 bff8e9fe0af",
16  "upgradeChannel": "Latest",
17  "primarySipUrl": "lys01-4-theapicave_rb@10.10.10.10",
18  "sipUrls": [
19    "lys01-4-theapicave_rb@10.10.10.10"
20  ],
21  "errorCodes": [],
22  "connectionStatus": "connected"}
```

# Onboarding API from On Prem to Cloud

- Before: Factory Reset, Physical Presence in Room to onboard to cloud.
- As of July 2019: Ability to insert an API command remotely with activation code to onboard directly even with a CUCM or Expressway connection active!

```
xcommand Webex Registration Start ?  
xCommand Webex Registration Start  
  ActivationCode(r): <S: 16, 19>  
  SecurityAction(r): <Harden, NoAction>
```



# 3<sup>rd</sup> Party Lights Reacting to Presence or Call

# In-Room Controls for Philips Hue

## Network Topology



Hue Bulb

---  
zigbee



172.17.101.71



172.17.101.83



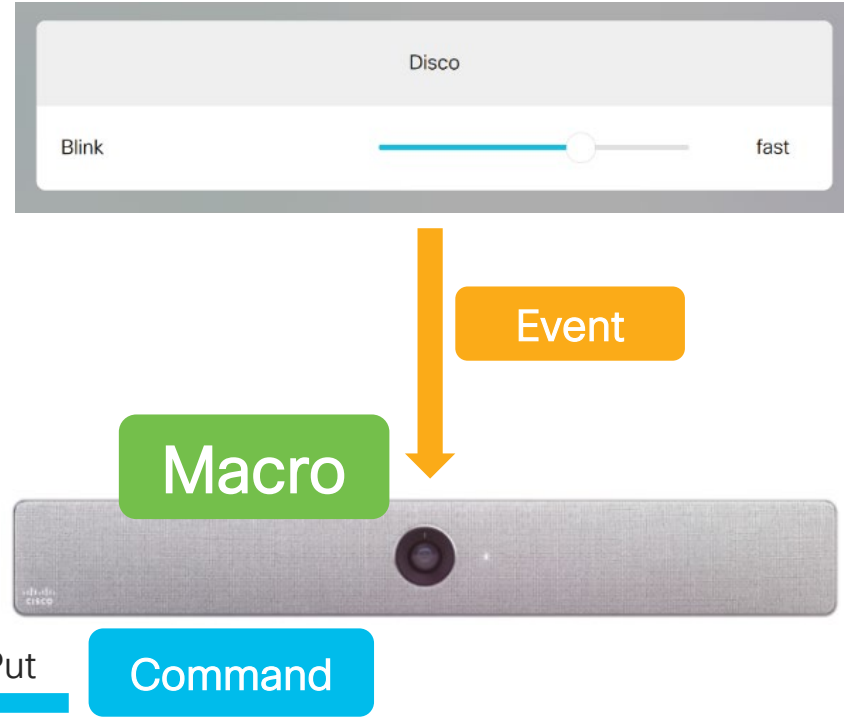
172.17.101.69

# Hue Controls

<http://bit.ly/ObjectIsAdvantag-HueDisco>



HTTP Put



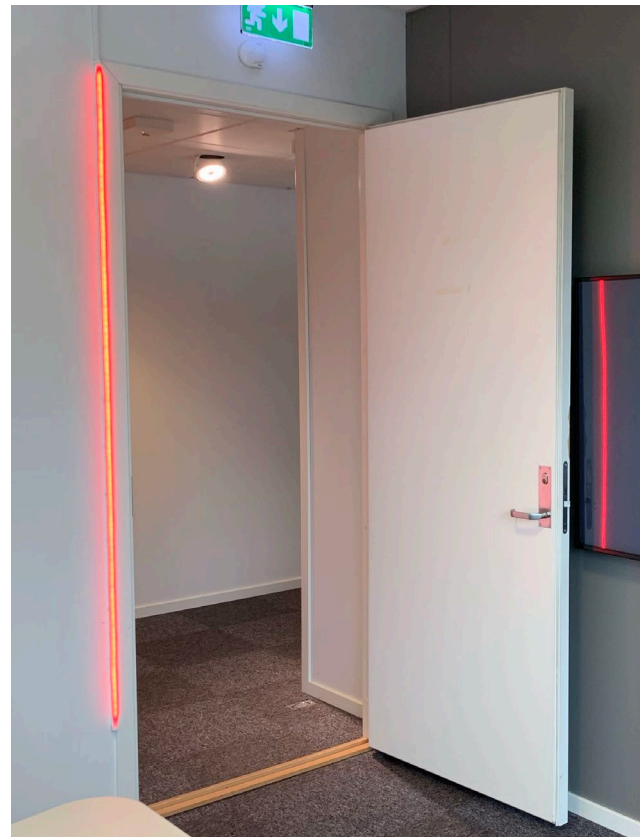
Free Room



Busy – Not In Call



Busy – In Call



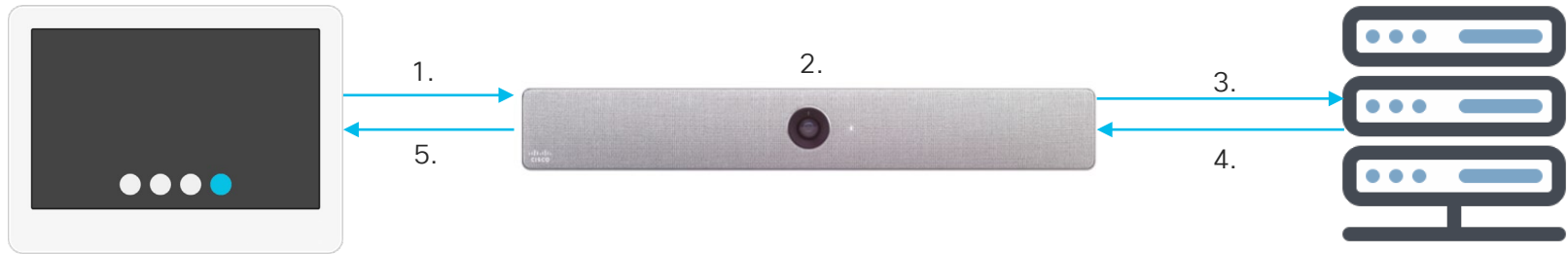




# Room Issues Feedback Example

# Flow Diagram

1. User Reports Problem from Touch 10
2. Macro picks up event from Touch 10
3. Macro Sends HTTP POST to 3rd party ticketing system
4. Ticketing System replies with response Payload
5. Reponse Payload is displayed to confirm to the end user confirming the reproting of the issue.



# In Room Control Designer

The screenshot displays the 'In-Room configuration' interface. On the left, a sidebar contains a 'Panels' section with a '+ Report Issue' button and an 'Always' status indicator. Below this is a 'Page content' section with three buttons: 'btn\_videoissue', 'btn\_audioissue', and 'btn\_messyroom', followed by a 'text' input field. The main workspace shows a configuration for the 'Report Issue' panel, which includes three buttons: 'Video Issue', 'Audio Issue', and 'Messy Room'. Below these buttons is a link: 'View Tickets at: <https://roomdevices.astzone.com/>'. The right sidebar contains a 'Widgets' section with a 'Panel is available:' dropdown set to 'Always'. Below this is an 'Icon' section with a grid of icons, where the 'Report Issue' icon (a circle with a question mark) is selected. At the bottom of the right sidebar is a 'Panel button color' section with a grid of 16 color swatches, where the yellow swatch is selected. A note below the color swatches reads: 'Note: This color is the system default for Files. Please choose the most similar activity to avoid a confusing user experience.' At the very bottom of the right sidebar is a 'Delete panel' button.

DEVNET-1462 © 2020 Cisco and/or its affiliates. All rights reserved. Cisco Public 43

# Listen to the Customised Panel

```
function feedbackWidget(event){
  if (event.WidgetId === 'btn_videoissue' && event.Type === 'clicked')
  {
    sendMonitoringUpdatePost("Video Issue Reported");

    xapi.command("UserInterface Message Alert Display", {
      Title: 'Feedback Recorded'
      , Text: 'Thank you for your feedback! We have opened a ticket! Have a great day!'
      , Duration: 3
    }).catch((error) => { console.error(error); });
  }
  else if (event.WidgetId === 'btn_audioissue' && event.Type === 'clicked')
  {
    sendMonitoringUpdatePost("Audio Issue Reported");

    xapi.command("UserInterface Message Alert Display", {
      Title: 'Feedback Recorded'
      , Text: 'Thank you for your feedback! We have opened a ticket! Have a great day!'
      , Duration: 3
    }).catch((error) => { console.error(error); });
  }
  else if (event.WidgetId === 'btn_messyroom' && event.Type === 'clicked')
  {
    sendMonitoringUpdatePost("Messy Room Reported");

    xapi.command("UserInterface Message Alert Display", {
      Title: 'Feedback Recorded'
      , Text: 'Thank you for your feedback! We have notified the cleaners! Have a great day!'
      , Duration: 3
    }).catch((error) => { console.error(error); });
  }
}
```

# Send HTTP Post

```
function sendMonitoringUpdatePost(message){  
    xapi.command('HttpClient Post', { AllowInsecureHTTPS: true, Header: 'Content-Type: application/json' , Url: MONITORING_URL}  
    , JSON.stringify(Object.assign({'message': message}, {'systemName': 'Cisco Feedback Demo Device'}))));  
}
```

# The RESULT!

## Device reports

Raw Data [Formatted Data](#)

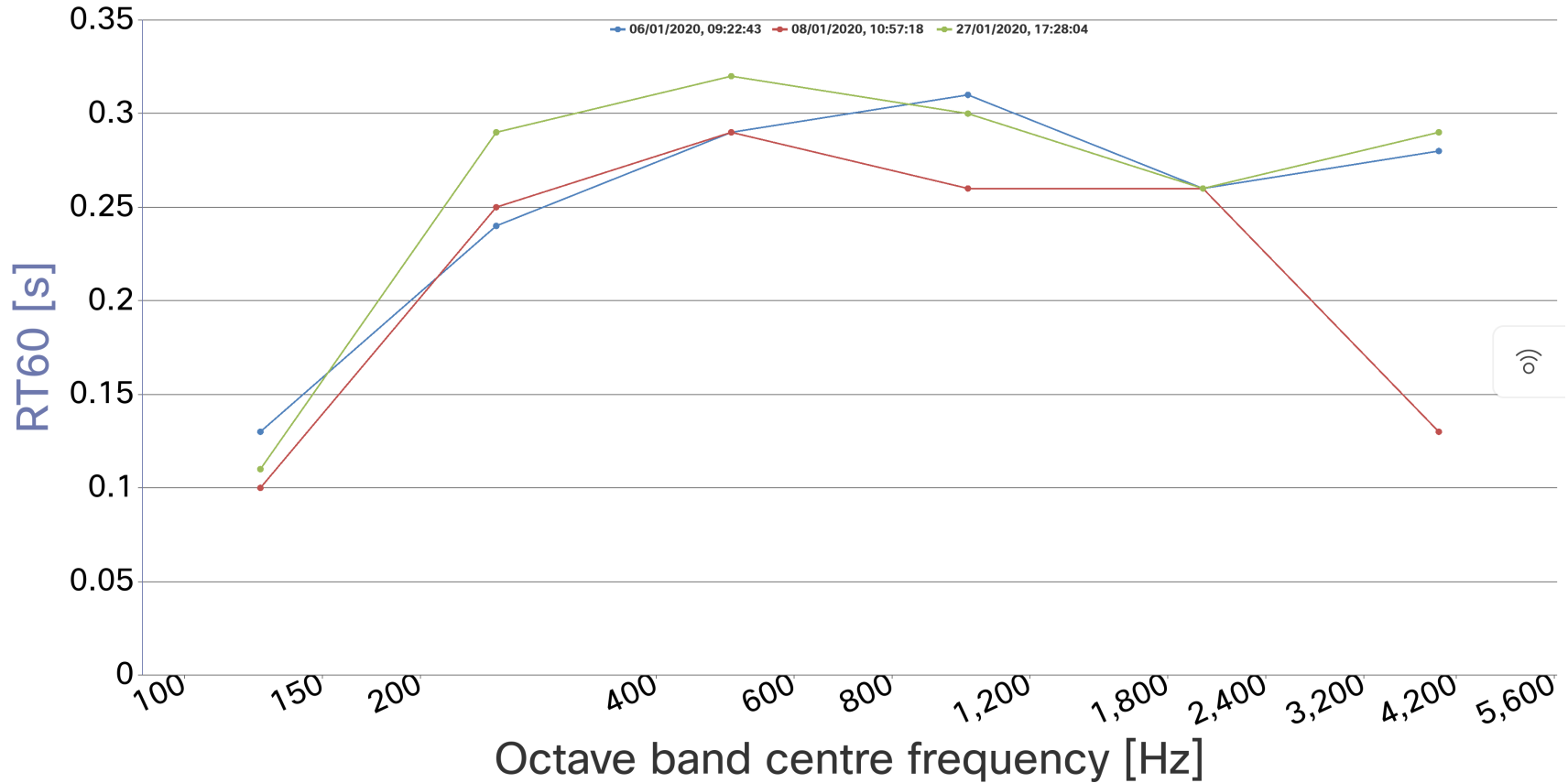
Case ID #	Timestamp	System Name	Message
1	01/05/2019 22:36:08	Cisco Live 2019 San Diego	Messy Room Reported

Clear data

# RT60 Reverb Macro (Quality Check Room Installs)

- We released a feature which allows you to run an Audio Diagnostics of the room to check for RT60 Levels!
- Using Macros we can send the results to a webpage and display the results on the screen!
  - But this could also be sent to a webserver and stored in a database! The power of HTTP POST!

# Reverberation time RT60



Trial Version

CanvasJS.com



```

1  const xapi = require('xapi');
2
3  function parseResult(data) {
4      let plotData = [];
5
6      for (let measurement of data["ReverberationTime"]) {
7          let octaveParts = measurement.Octave.split(" ");
8          let x = parseInt(octaveParts[0]) * (octaveParts[1] === "kHz" ? 1000 : 1);
9          let y = parseFloat(measurement.RT60.split(" ")[0]);
10         plotData.push({x,y});
11     }
12
13     return plotData;
14 }
15
16 function openPlot(data) {
17     const plotData = parseResult(data);
18     xapi.command("UserInterface WebView Display", { 'Url': 'http://<host>.com/plot.html?data=' +
19         JSON.stringify(plotData), 'Title': 'RT60 results' });
20 }
21
22 xapi.event.on("UserInterface/Extensions/Panel/Clicked", (event) => {
23     if (event.PanelId !== "panel_sound_check")
24         return;
25
26     xapi.command("Audio Diagnostics Advanced Run", {MeasurementLength: 1, Volume: 10})
27     .then((result) => {
28         openPlot(result);
29     });
30 });

```

# What's Next in 2020?

# Format enhancements for JSXAPI and Macros

- JSXAPI 5.0 (Just Released!)
  - Ported to TypeScript to enable more friendly code writing and reading! (BACKWARDS COMPATIBLE!)
- Macros TypeScript Support Coming Soon!

```
// Set up a call
xapi.command('Dial', { Number: 'user@example.com' });

// Fetch volume and print it
xapi.status
  .get('Audio Volume')
  .then((volume) => { console.log(volume); });

// Set a configuration
xapi.config.set('SystemUnit Name', 'My System');

// Listen to feedback
const off = xapi.event.on('Standby', (event) => {
  // ...
});

// De-register feedback
off();
```

OLD

```
// Set up a call
xapi.Command.Dial({ Number: 'user@example.com' });

// Fetch volume and print it
xapi.Status.Audio.Volume
  .get()
  .then((volume) => { console.log(volume); });

// Set a configuration
xapi.Config.SystemUnit.Name.set('My System');

// Listen to feedback
const off = xapi.Event.Standby.on((event) => {
  // ...
});

// De-register feedback
off();
```

NEW

# Configuration APIs for REST (Cloud / Hybrid)

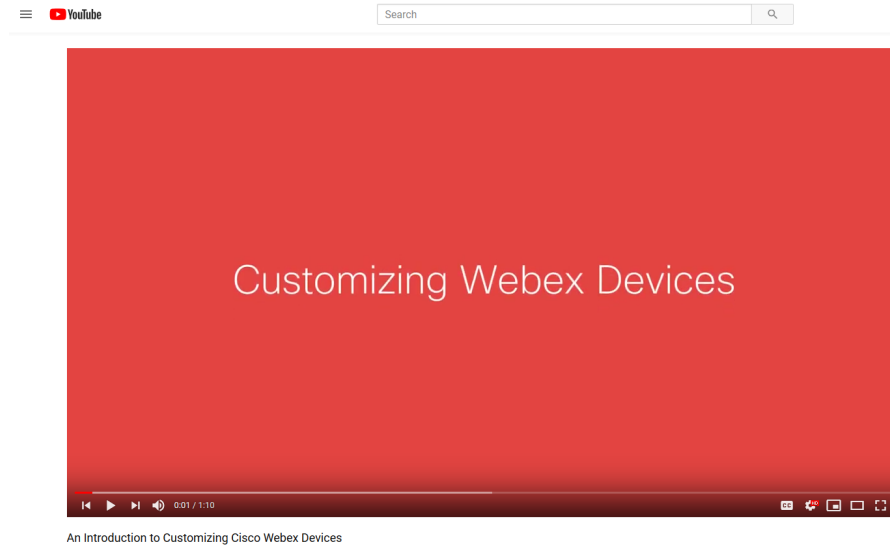
- Ability to edit configurations on any device using the authorizations model of the Webex Platform
- Works for RoomOS Devices as well as Share!
- Example
  - <https://api.ciscospark.com/v1/device/{DeviceID}/config/audio/defaultvolume/> (WIP, may change)

# Ability to make Hidden Panels for UI Customization

- Already editable via the API!
- xCommand UserInterface Extensions Panel Update
  - Color: <S: 0, 255>
  - Icon: <Briefing, Camera, Concierge, Disc, Handset, Help, Helpdesk, Home, Hvac, Info, Input, Language, Laptop, Lightbulb, Proximity, Record, Spark, Tv, Webex, General>
  - Name: <S: 0, 255>
  - PanelId(r): <S: 0, 255>
  - **Visibility: <Auto, Hidden>**
- Editor setting coming soon!

# How to continue learning?

- [https://www.youtube.com/playlist?list=PL\\_YnWo4XhzTe8CyOppyhheB8UN40mDYiX](https://www.youtube.com/playlist?list=PL_YnWo4XhzTe8CyOppyhheB8UN40mDYiX)



# Resources

- UI Extensions Learning Lab
  - <https://learninglabs.cisco.com/lab/collab-xapi-controls/step/1>
- xAPI samples
  - <https://github.com/ObjectIsAdvantag/xapi-samples>
  - <https://github.com/CiscoDevNet/roomdevices-macros-samples/>
  - <http://cs.co/roomdevices>
- ‘xAPI devs’ community space
  - <https://eurl.io/#rkp76XDrG>
- awesome-xapi: curated community resources
  - <https://github.com/CiscoDevNet/awesome-xapi>

# Complete your online session survey



- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on [ciscolive.com/emea](https://ciscolive.com/emea).

Cisco Live sessions will be available for viewing on demand after the event at [ciscolive.com](https://ciscolive.com).



# Continue your education



Demos in the  
Cisco campus



Walk-in  
self-paced labs



Meet the engineer  
1:1 meetings



Related sessions



Thank you





You make **possible**