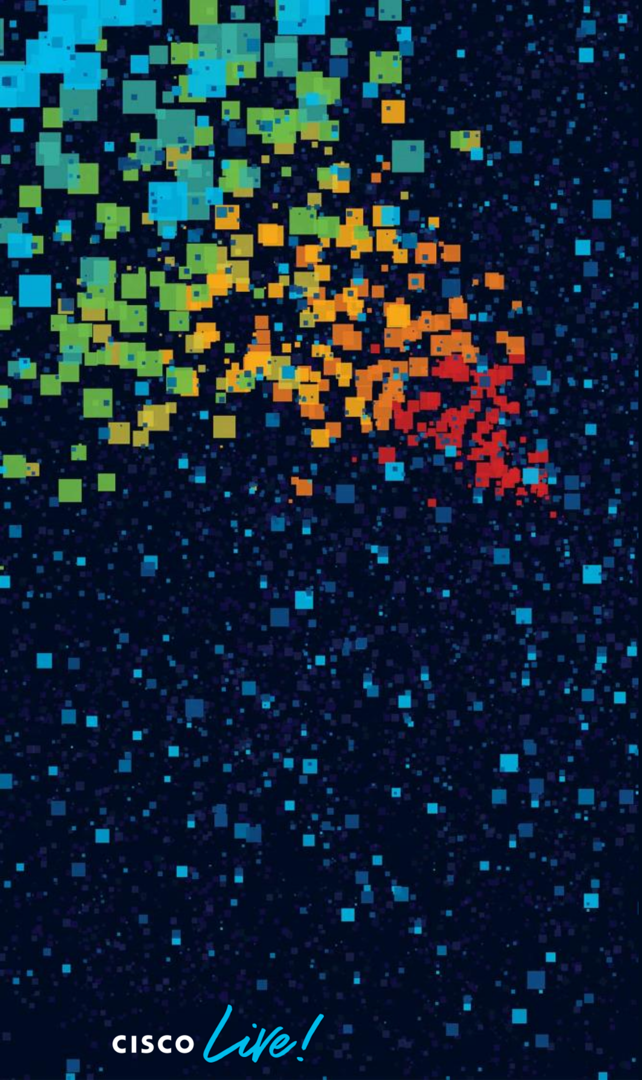
Possibilities

#CiscoLive

Advanced Coding and Deployment for Cisco Video Devices

Stève Sfartz, stsfartz@cisco.com
Principal Architect
Cisco DevNet, Customer Experience

DGTL-BRKPRG-3244



Agenda

- CE Programmability
- Cloud Programmability

/Cisco/CX/DevNet/StèveSfartz

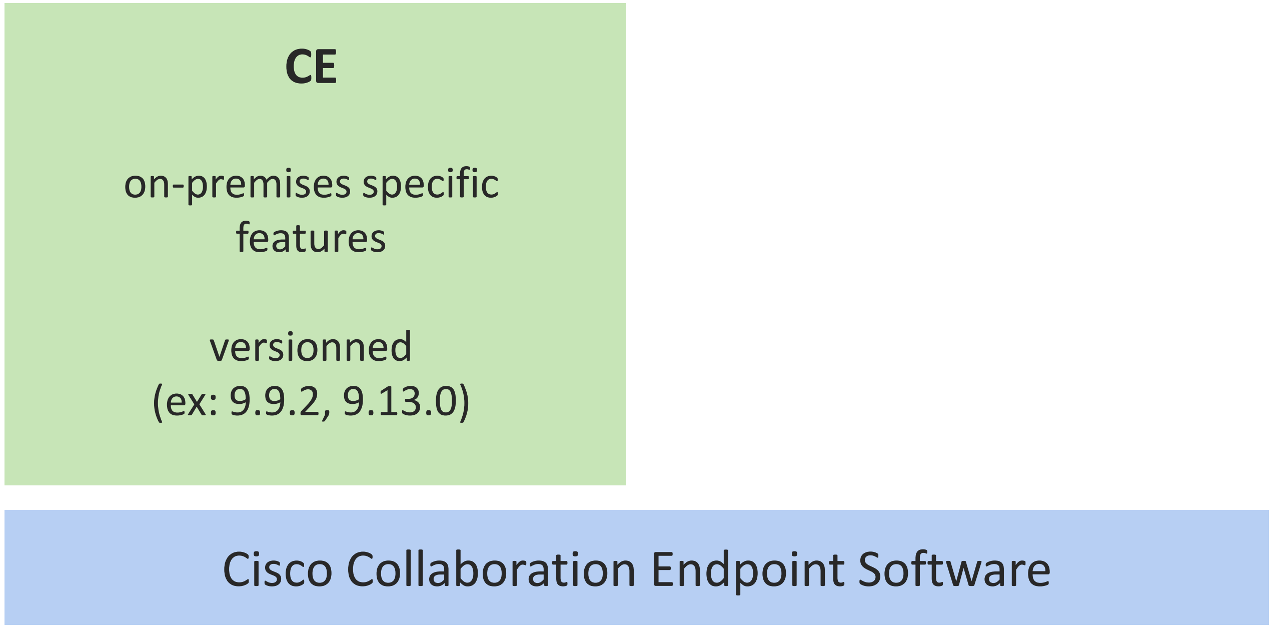
- API Architect at Cisco - Customer eXperience
- Working to deliver the greatest API and developer eXperience for our customers & partners
- Technical lead for Cisco API Experience group and API Style Guide
- Subject Matter Expert for Webex APIs
- Contributor to DevNet CodeExchange and AutomationExchange

webex: stsfartz@cisco.com
github: ObjectIsAdvantag
twitter: @SteveSfartz

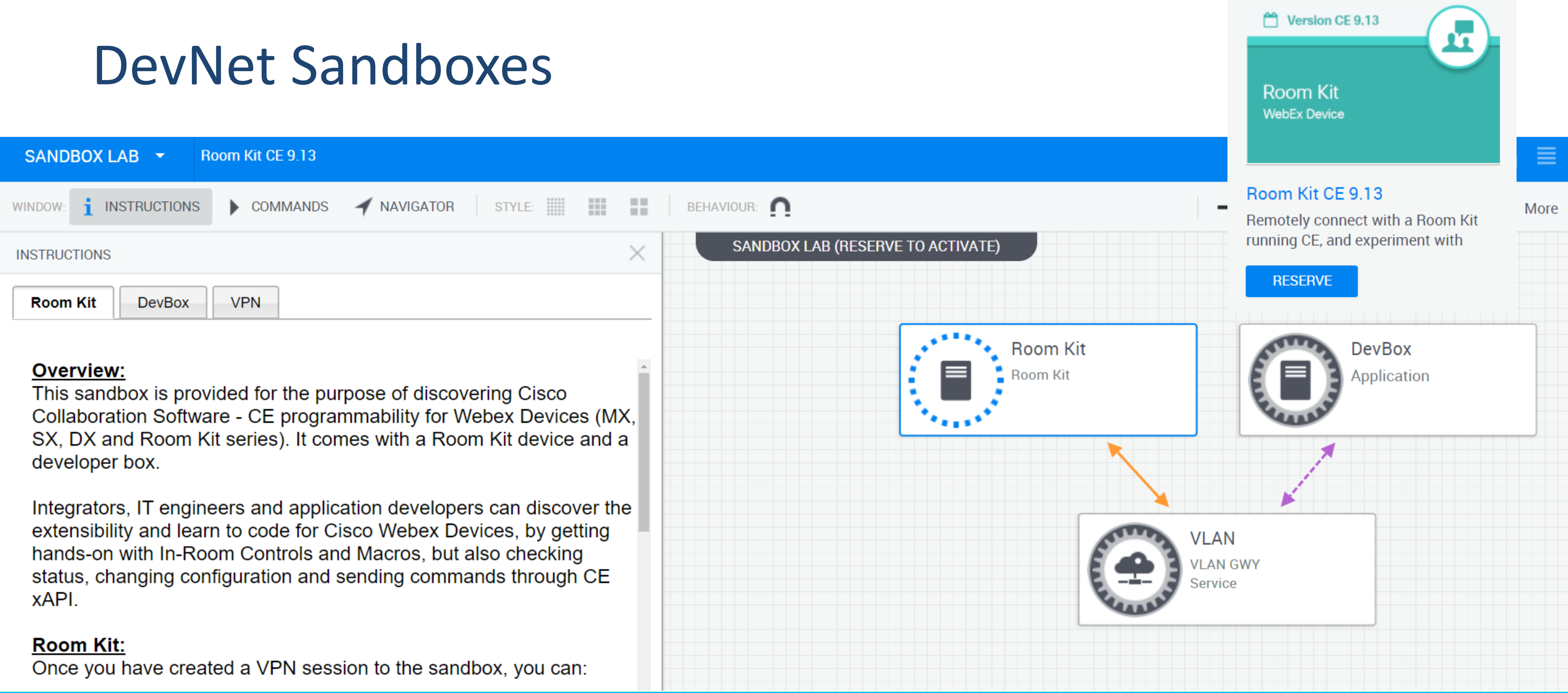
“vision without execution is hallucination”

Chapter I
CE Programmability

CE and RoomOS


CE
on-premises specific features
versionned
(ex: 9.9.2, 9.13.0)
Cisco Collaboration Endpoint Software

DevNet Sandboxes



SANDBOX LAB Room Kit CE 9.13

WINDOW: INSTRUCTIONS COMMANDS NAVIGATOR STYLE: BEHAVIOUR: More

INSTRUCTIONS

Room Kit DevBox VPN

Overview:
This sandbox is provided for the purpose of discovering Cisco Collaboration Software - CE programmability for Webex Devices (MX, SX, DX and Room Kit series). It comes with a Room Kit device and a developer box.

Integrators, IT engineers and application developers can discover the extensibility and learn to code for Cisco Webex Devices, by getting hands-on with In-Room Controls and Macros, but also checking status, changing configuration and sending commands through CE xAPI.

Room Kit:
Once you have created a VPN session to the sandbox, you can:

Room Kit CE 9.13

Remotely connect with a Room Kit running CE, and experiment with

RESERVE

https://developer.cisco.com/sandbox/

Cisco Collaboration Devices Programmability

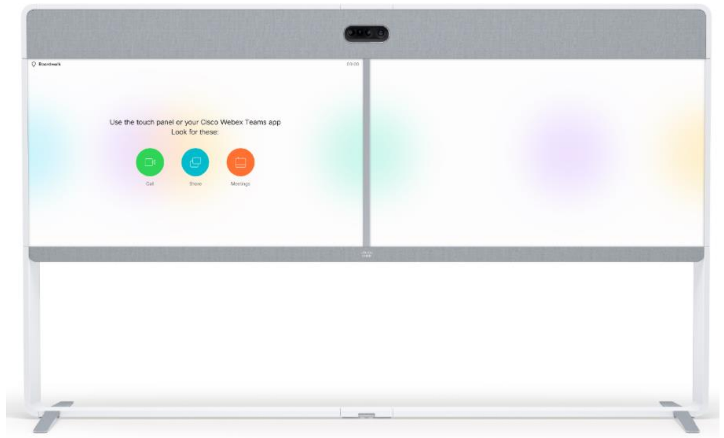
Available on all devices running CE & RoomOS

RoomKit standard, pro, mini

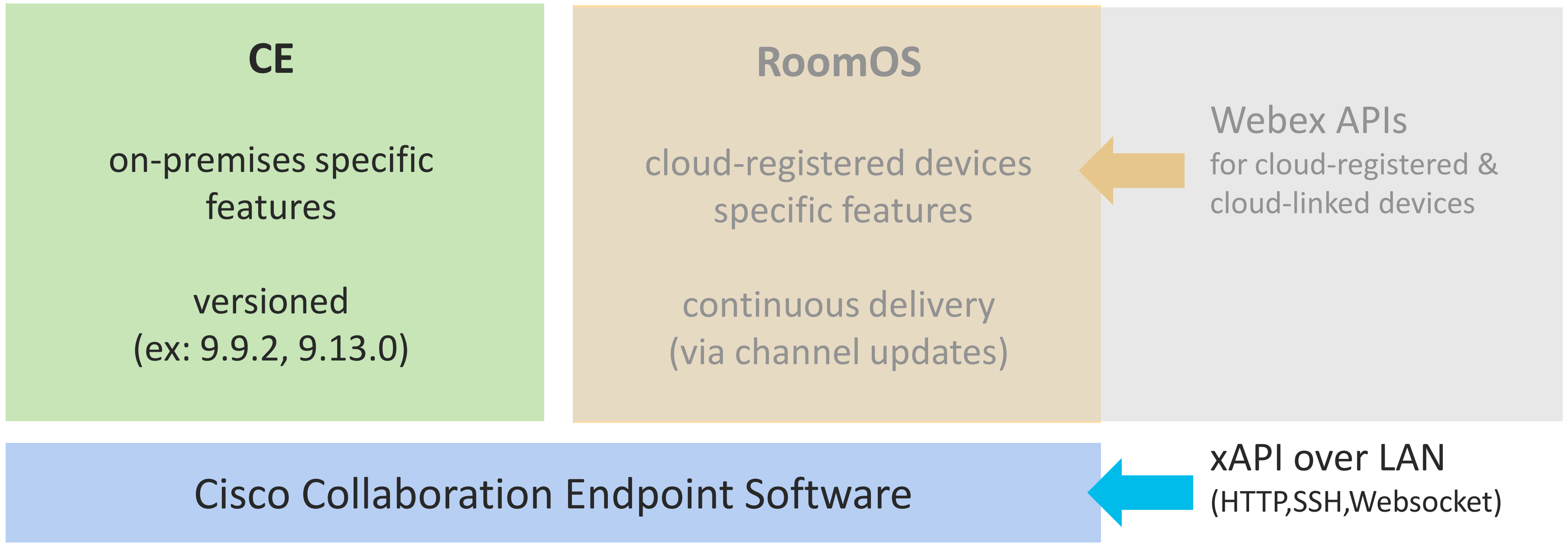
*no Macros on SX10

Webex Board

DX, DeskPro

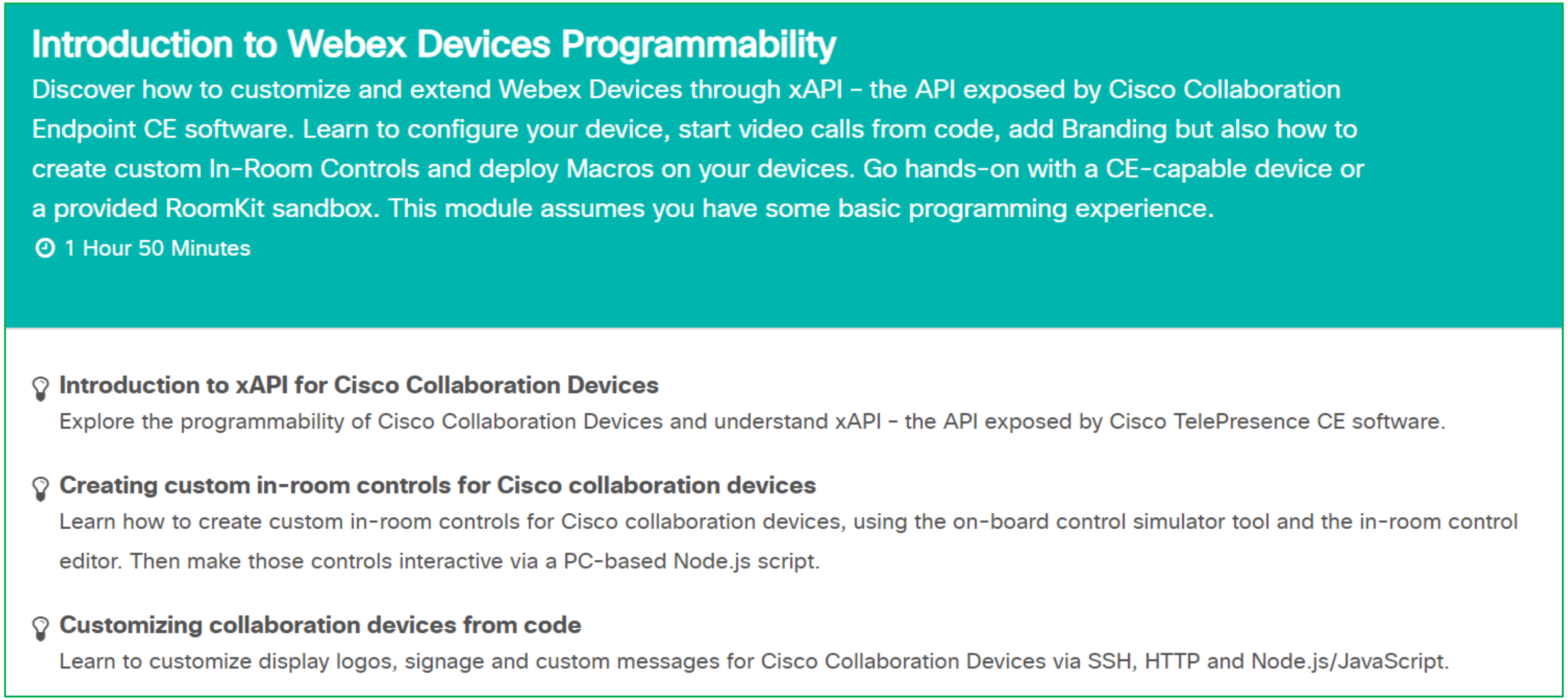


CE and RoomOS


CE
on-premises specific features
versioned
(ex: 9.9.2, 9.13.0)
RoomOS
cloud-registered devices specific features
continuous delivery
(via channel updates)
Webex APIs
for cloud-registered & cloud-linked devices
Cisco Collaboration Endpoint Software
xAPI over LAN
(HTTP,SSH,Websocket)

xAPI module at DevNet

https://developer.cisco.com/learning/modules/xapi-intro



Introduction to Webex Devices Programmability

Discover how to customize and extend Webex Devices through xAPI – the API exposed by Cisco Collaboration Endpoint CE software. Learn to configure your device, start video calls from code, add Branding but also how to create custom In-Room Controls and deploy Macros on your devices. Go hands-on with a CE-capable device or a provided RoomKit sandbox. This module assumes you have some basic programming experience.

1 Hour 50 Minutes

Introduction to xAPI for Cisco Collaboration Devices

Explore the programmability of Cisco Collaboration Devices and understand xAPI - the API exposed by Cisco TelePresence CE software.

Creating custom in-room controls for Cisco collaboration devices

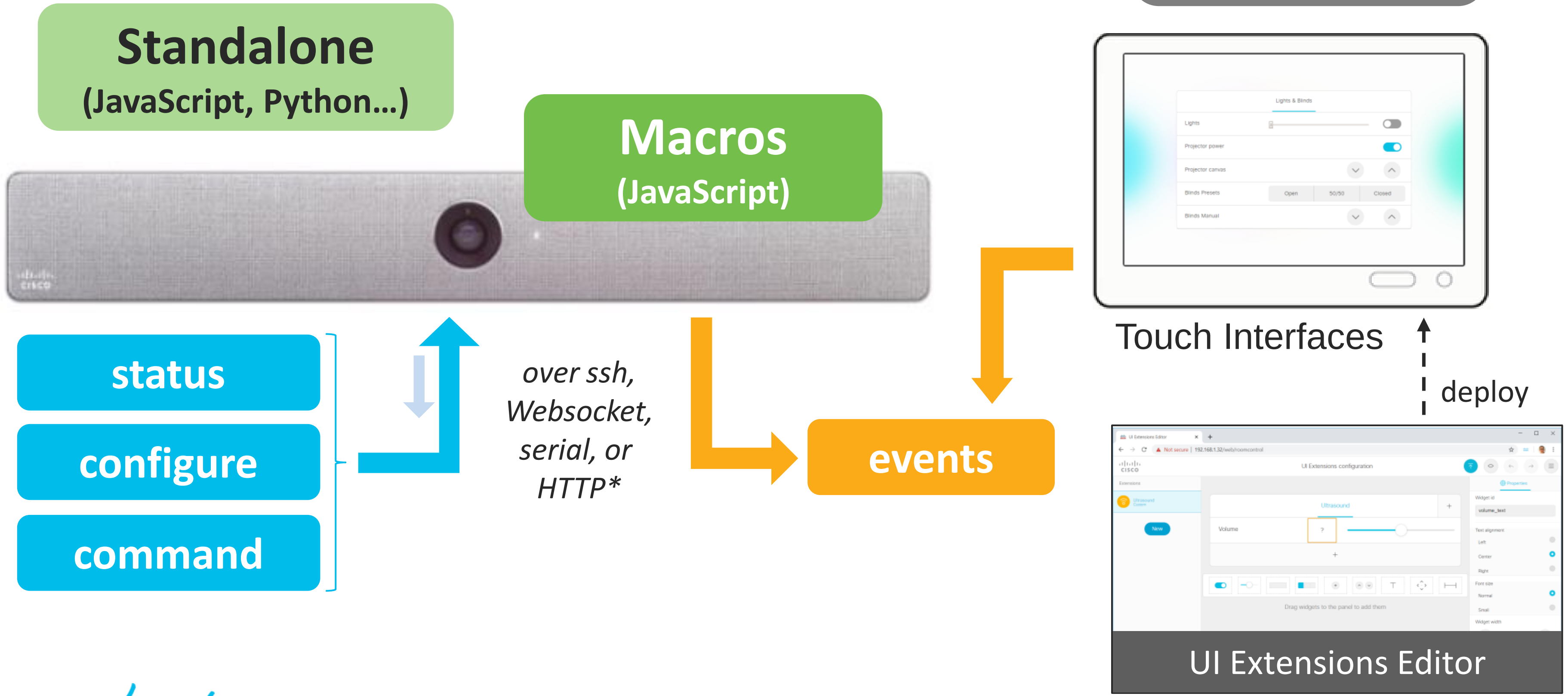
Learn how to create custom in-room controls for Cisco collaboration devices, using the on-board control simulator tool and the in-room control editor. Then make those controls interactive via a PC-based Node.js script.

Customizing collaboration devices from code

Learn to customize display logos, signage and custom messages for Cisco Collaboration Devices via SSH, HTTP and Node.js/JavaScript.

CE Programmability (xAPI)

Detailed in BRKPRG-3244 CL20B



Detailed in BRKPRG-3244 CL20B

Is the device configured for HttpClient?

```
// The device must be configured to support HttpClient
//   - xConfiguration HttpClient Mode: On
//   - xConfiguration HttpClient AllowInsecureHTTPS: True
```

Is this the correct event?

Is the panel deployed?

How can I pass these values?

What if the request fails?

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

xapi.event.on('UserInterface Extensions Widget Action', (event) => {
  if ((event.WidgetId == 'BRKDEV_slider') && (event.Type == 'released')) {
    // Change color for Philips Hue light
    const HUE_BRIDGE = '192.168.1.33';
    const HUE_USERNAME = 'EM2Vg2GtNUqAASukv47wm1pWY0FayFe48D03f6Cb';
    const HUE_LIGHT = 1; // number of the light in your deployment

    xapi.command('HttpClient Put', {
          Header: ["Content-Type: application/json"],
          Url: `http://${HUE_BRIDGE}/api/${HUE_USERNAME}/lights/${HUE_LIGHT}/state`,
          AllowInsecureHTTPS: "True",
          ResultBody: 'plaintext'
        },
        JSON.stringify({ "hue": Math.round(event.Value / 255 * 65535), "sat": 255 }));
  }
```

Macro Runtime

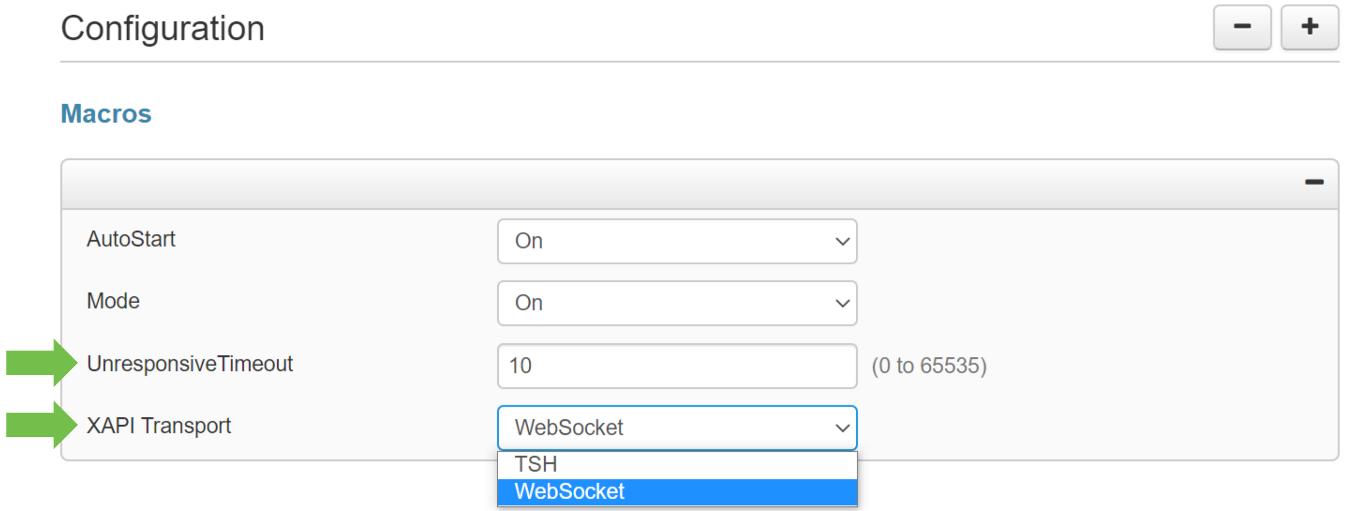
- Available for CE & RoomOS Webex devices (except SX10)
- Duktape JavaScript runtime with Babel
  - babel-preset-latest' is specified to support latest ES6+ features
  - async/await is supported
- Macros are executed as 'Admin' role

new

- Increased number of Macros (30 and more, depends on device capacity)
- Maximum of 10 active macros
- Macro tutorial accessible from the codec
  - http://<ip-address>/static/docs/macro-tutorial.pdf

Tuning the Macro Runtime

xAPI transport now defaults to WebSocket



Configuration

Macros

AutoStart	On
Mode	On
UnresponsiveTimeout	10 (0 to 65535)
XAPI Transport	WebSocket

Node.js ‘jsxapi’ module

https://github.com/cisco-ce/jsxapi

- Connect via ‘ssh’ or ‘websockets’ (since jsxapi v4.4 & CE 9.8)

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

jsxapi
  .connect('ssh://host.example.com', {
    username: 'admin',
    password: 'password',
  })
  .on('error', console.error)
  .on('ready', async (xapi) => {
    const volume = await xapi.status.get('Audio Volume');
    console.log(`volume is: ${volume}`);
    xapi.close();
  });
```

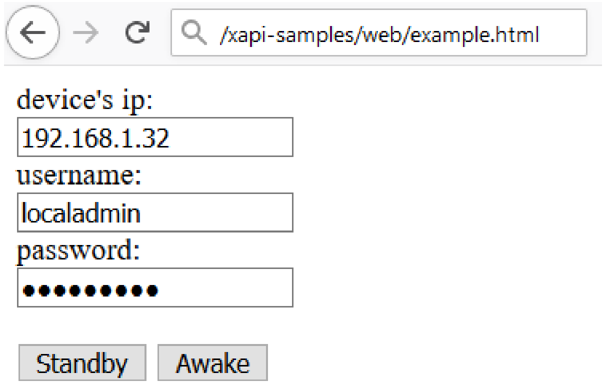
```
jsxapi
  .connect('wss://host.example.com', {
    username: 'admin',
    password: 'password',
  })
```

Benefits
- Better performances
- Manage CE directly from Web Applications

Manage Devices from Web Applications

https://github.com/CiscoDevNet/xapi-samples/blob/master/web/example.html

```
<script src="./jsxapi-501.min.js"></script>
<script>
   let xapi = jsxapi.connect(`wss://${document.device.ip.value}`,
      { username: document.device.user.value, password: document.device.passwd.value });
   xapi.on('ready', () => {
      console.log("connected to device, ready to send xAPI requests");
   });

   function standby() {
      xapi.command("Standby Activate")
         .then(() => console.log("device in Standby mode"))
         .catch((err) => console.log(`error: ${err.message}`));
   }

   function awake() {
      xapi.command("Standby Deactivate")
         .then(() => console.log("device is now Active"))
         .catch((err) => console.log(`error: ${err.message}`));
   }

</script>
```


/xapi-samples/web/example.html

device's ip: 192.168.1.32

username: localadmin

password: ●●●●●●●●●●

Enhanced coding style

Available for both Macro runtime and 'jsxapi' module

Invoking a command:

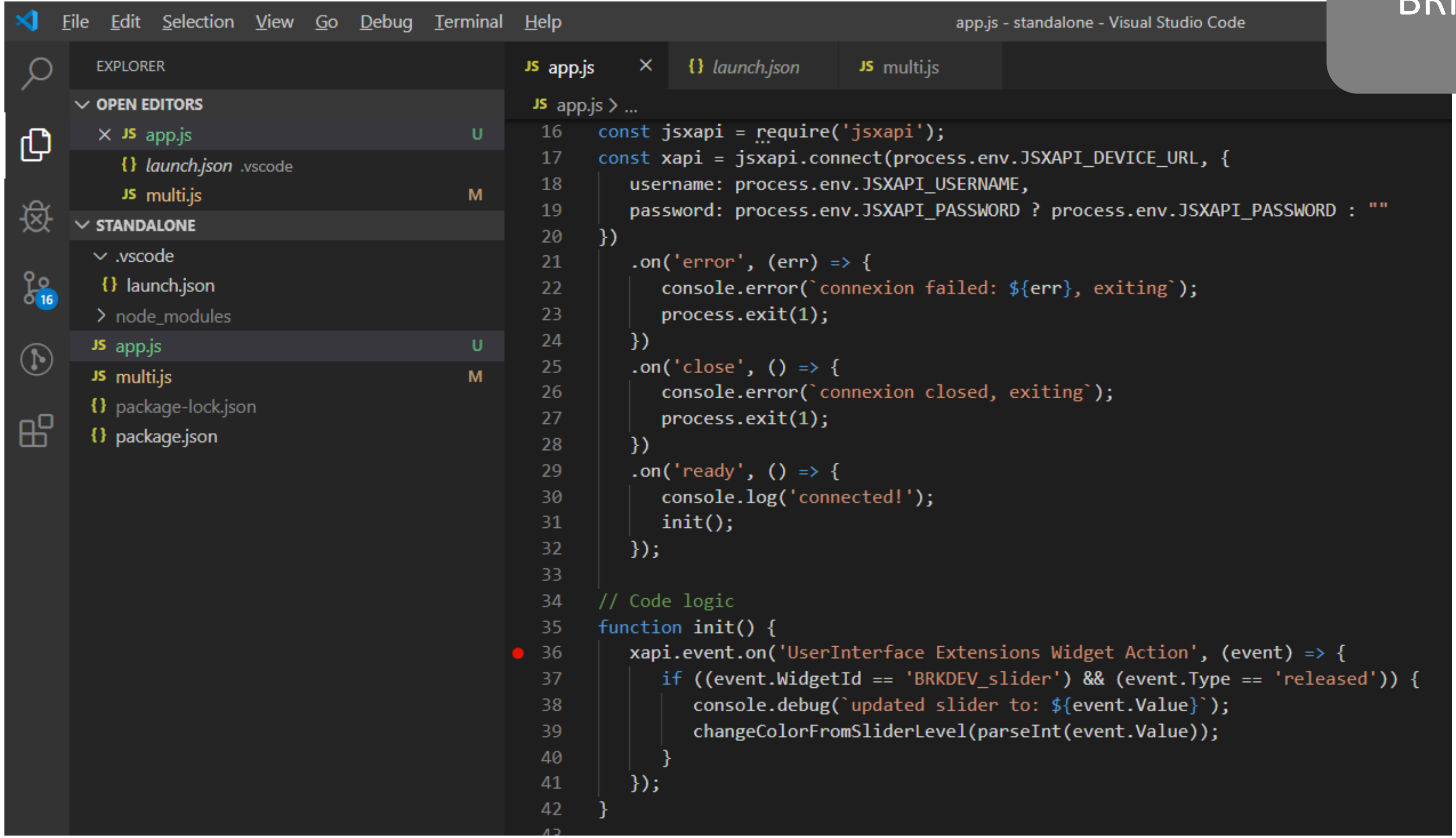
```
xapi.Command.Dial({ Number: 'user@example.com' });
```

Getting a status:

```
xapi.Status.Audio.Volume.get().then((volume) => {
  console.log(volume);
});
```

IDE: a better coding experience!

Detailed in BRKPRG-3244 CL20B

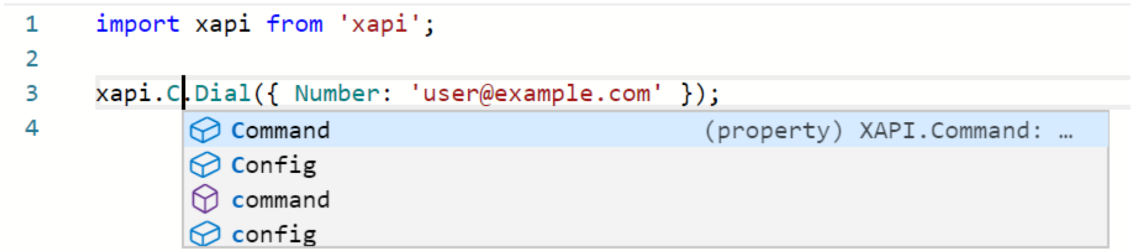


```javascript
const jsxapi = require('jsxapi');
const xapi = jsxapi.connect(process.env.JSXAPI_DEVICE_URL, {
  username: process.env.JSXAPI_USERNAME,
  password: process.env.JSXAPI_PASSWORD ? process.env.JSXAPI_PASSWORD : ""
})
  .on('error', (err) => {
    console.error(`connexion failed: ${err}, exiting`);
    process.exit(1);
  })
  .on('close', () => {
    console.error(`connexion closed, exiting`);
    process.exit(1);
  })
  .on('ready', () => {
    console.log('connected!');
    init();
  });

// Code logic
function init() {
  xapi.event.on('UserInterface Extensions Widget Action', (event) => {
    if ((event.WidgetId == 'BRKDEV_slider') && (event.Type == 'released')) {
      console.debug(`updated slider to: ${event.Value}`);
      changeColorFromSliderLevel(parseInt(event.Value));
    }
  });
}
```

Enhanced Macro Editor

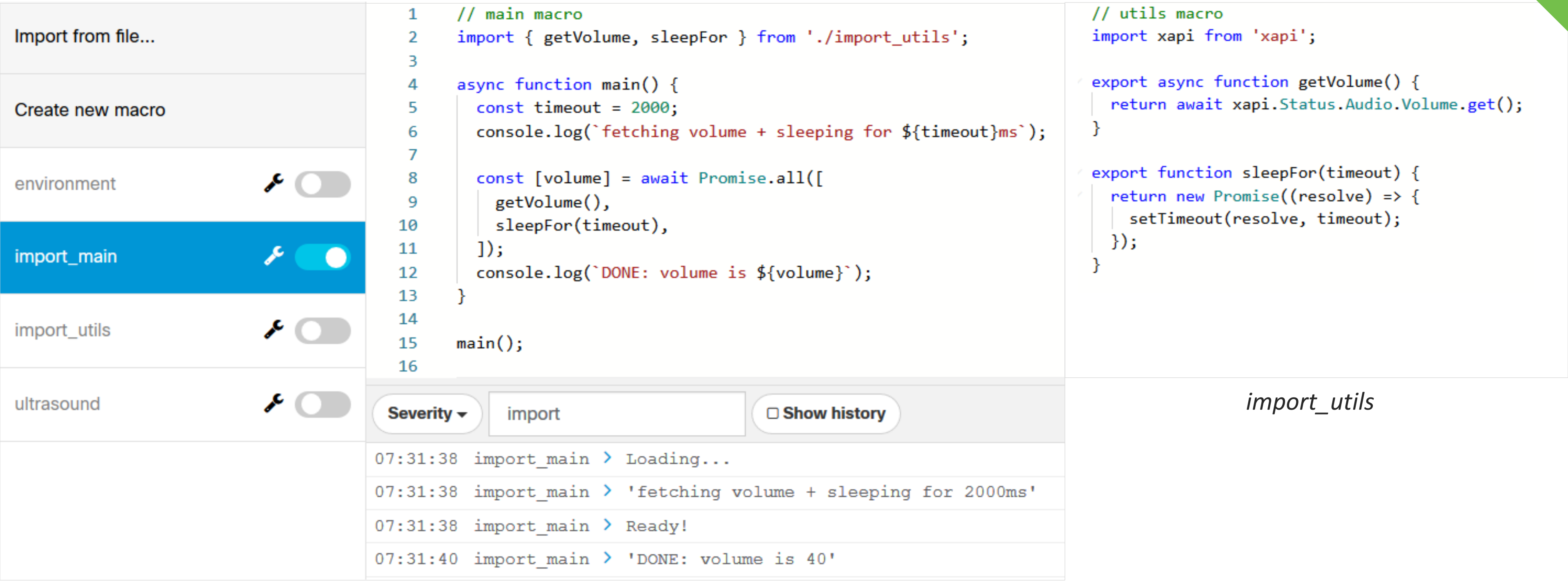
new

- The Macro Editor now leverages ‘Monaco’
- ‘Monaco’ is the code editor that powers Visual Studio Code
- Better code completion and warnings



```
import xapi from 'xapi';

xapi.C.Dial({ Number: 'user@example.com' });

```

Reuse code across Macros



```
// main macro
import { getVolume, sleepFor } from './import_utils';

async function main() {
  const timeout = 2000;
  console.log(`fetching volume + sleeping for ${timeout}ms`);

  const [volume] = await Promise.all([
    getVolume(),
    sleepFor(timeout),
  ]);
  console.log(`DONE: volume is ${volume}`);
}

main();
```

```
07:31:38  import_main > Loading...
07:31:38  import_main > 'fetching volume + sleeping for 2000ms'
07:31:38  import_main > Ready!
07:31:40  import_main > 'DONE: volume is 40'
```

```
// utils macro
import xapi from 'xapi';

export async function getVolume() {
  return await xapi.Status.Audio.Volume.get();
}

export function sleepFor(timeout) {
  return new Promise((resolve) => {
    setTimeout(resolve, timeout);
  });
}
```

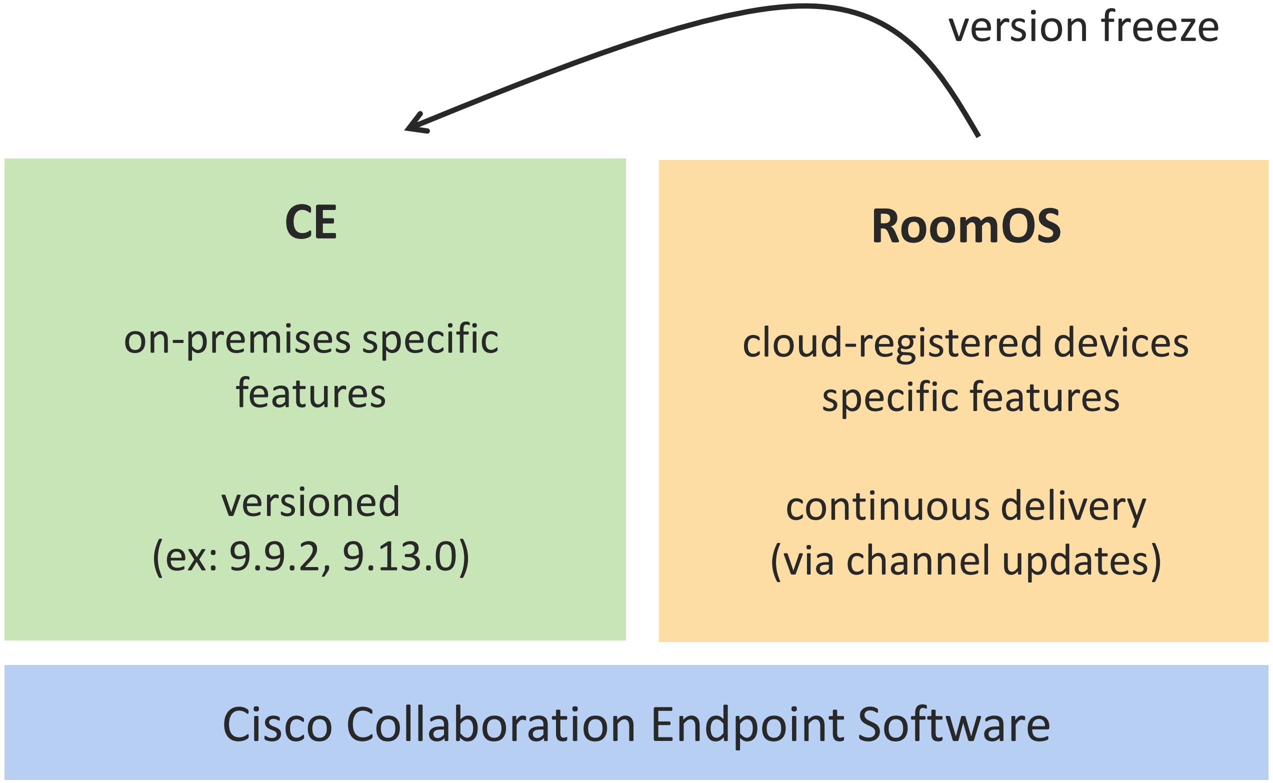
import_utils

What's new for CE Programmability

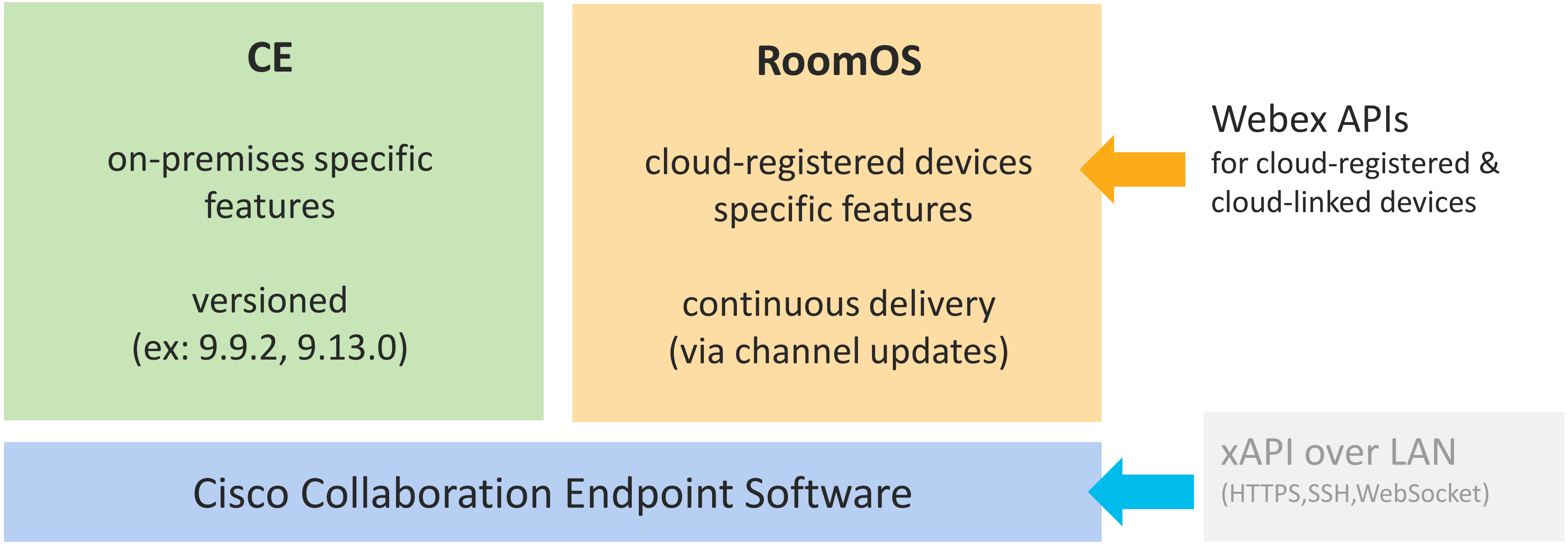
- Enhanced Macro Editor
- Macro Runtime
  - Transport defaults to Websocket
  - Increased number of Macros
  - Reuse code across Macros

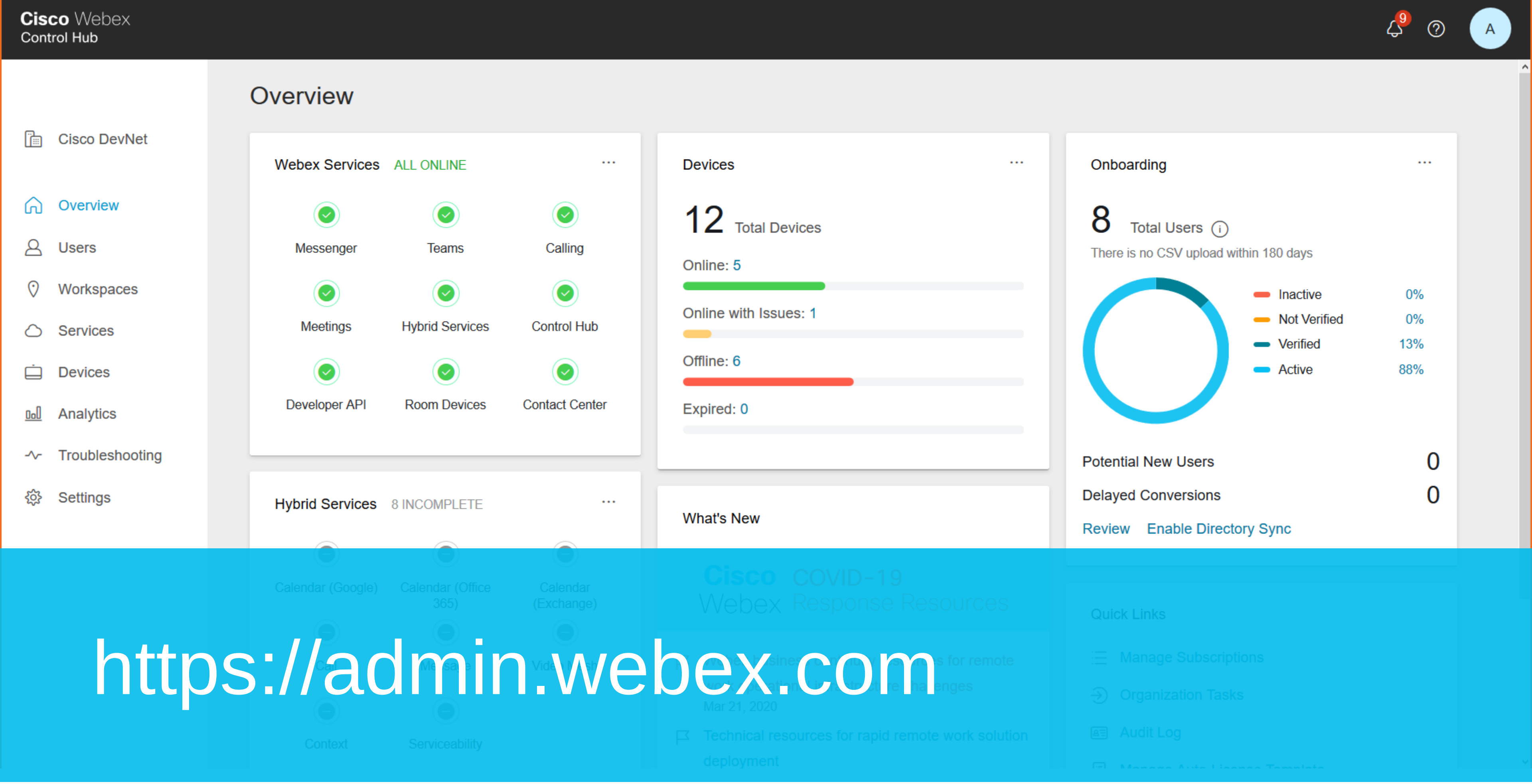
Chapter II
Cloud Programmability for Webex Devices

CE and RoomOS


version freeze
CE
on-premises specific features
versioned
(ex: 9.9.2, 9.13.0)
RoomOS
cloud-registered devices specific features
continuous delivery
(via channel updates)
Cisco Collaboration Endpoint Software

CE and RoomOS


CE
on-premises specific features
versioned (ex: 9.9.2, 9.13.0)
RoomOS
cloud-registered devices specific features
continuous delivery (via channel updates)
Webex APIs
for cloud-registered & cloud-linked devices
Cisco Collaboration Endpoint Software
xAPI over LAN
(HTTPS,SSH,WebSocket)


Cisco Webex
Control Hub

Overview

Cisco DevNet

- Overview
- Users
- Workspaces
- Services
- Devices
- Analytics
- Troubleshooting
- Settings

Webex Services ALL ONLINE

Messenger, Teams, Calling, Meetings, Hybrid Services, Control Hub, Developer API, Room Devices, Contact Center

Devices

12 Total Devices

Online: 5

Online with Issues: 1

Offline: 6

Expired: 0

Onboarding

8 Total Users

There is no CSV upload within 180 days

Inactive	0%
Not Verified	0%
Verified	13%
Active	88%

Potential New Users	0
Delayed Conversions	0

Review Enable Directory Sync

Hybrid Services 8 INCOMPLETE

Calendar (Google), Calendar (Office 365), Calendar (Exchange), Context, Serviceability

What's New

Cisco COVID-19 Webex Response Resources

Mar 21, 2020

Technical resources for rapid remote work solution deployment

Quick Links

Manage Subscriptions

Organization Tasks

Audit Log

https://admin.webex.com

Webex APIs for Devices

coming

/devices	/places	/xapi	/device Configurations	/workspaces
GET List Devices	GET List Places	GET Query Status	GET List Configurations	GET List Workspaces
POST Activation code	POST Create Place	POST Execute Command		POST Create Workspace
GET Get details	GET Get details			GET Get details
	PUT Update Place		PATCH Update Configurations	PUT Update Workspace
DELETE Delete Device	DELETE Delete Place			DELETE Delete Workspace

Managing Devices & Places

https://developer.webex.com/docs/api/v1/devices

Detailed in DGTL-DEVNET-2610

/devices	/places
GET List Devices	GET List Places
POST Activation code	POST Create a Place
GET Get details	GET Get details
	PUT Update a Place
DELETE Delete a Device	DELETE Delete a Place

<u>OAuth scopes</u>

for Webex administrators:

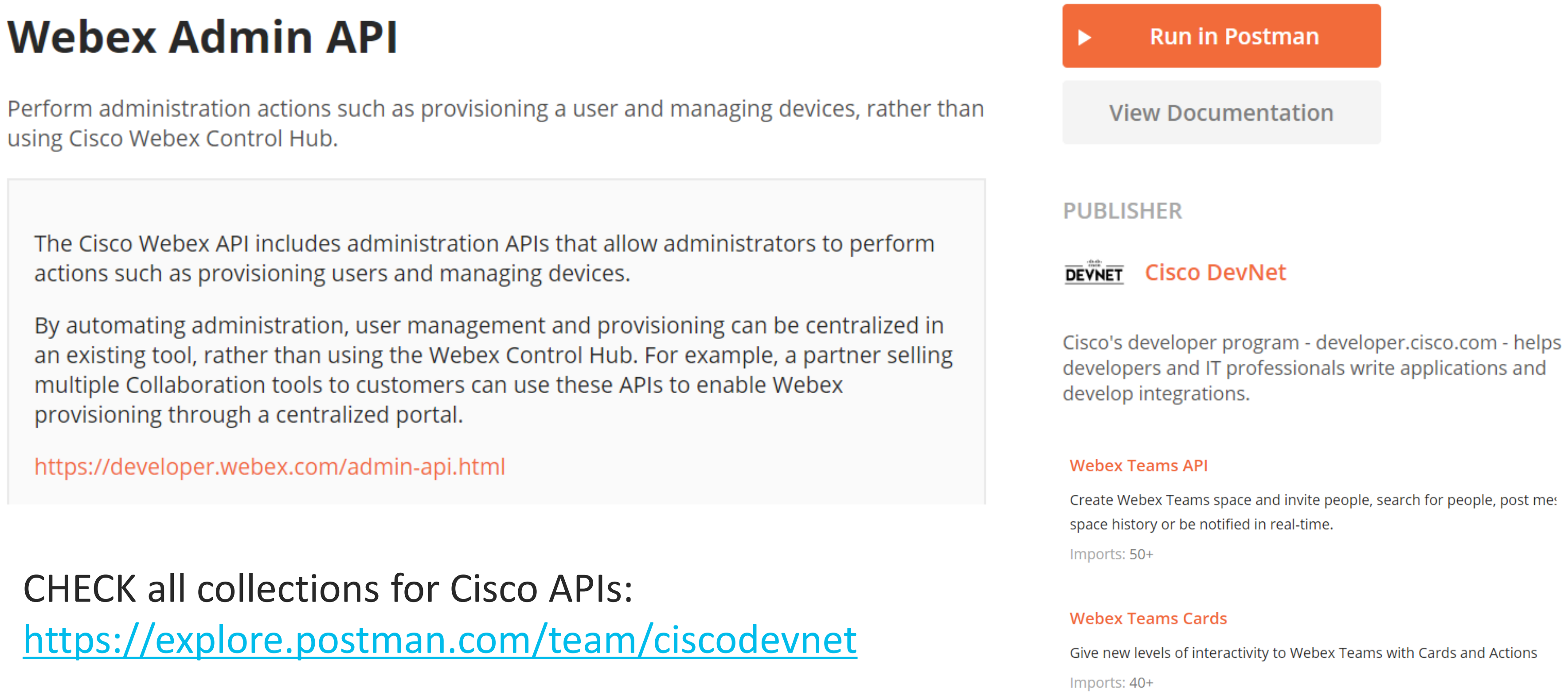
- spark-admin:devices_read
- spark-admin:devices_write
- spark-admin:places_read
- spark-admin:places_write
- identity:placeonetimepassword_create

for Webex users:

- spark:devices_read
- spark:devices_write
- spark:places_read
- spark:places_write

Managing Devices and Places [demo]

Postman collections for Webex



Webex Admin API

Perform administration actions such as provisioning a user and managing devices, rather than using Cisco Webex Control Hub.

The Cisco Webex API includes administration APIs that allow administrators to perform actions such as provisioning users and managing devices.

By automating administration, user management and provisioning can be centralized in an existing tool, rather than using the Webex Control Hub. For example, a partner selling multiple Collaboration tools to customers can use these APIs to enable Webex provisioning through a centralized portal.

https://developer.webex.com/admin-api.html

PUBLISHER

Cisco DevNet

Cisco's developer program - developer.cisco.com - helps developers and IT professionals write applications and develop integrations.

Webex Teams API

Create Webex Teams space and invite people, search for people, post mes
space history or be notified in real-time.

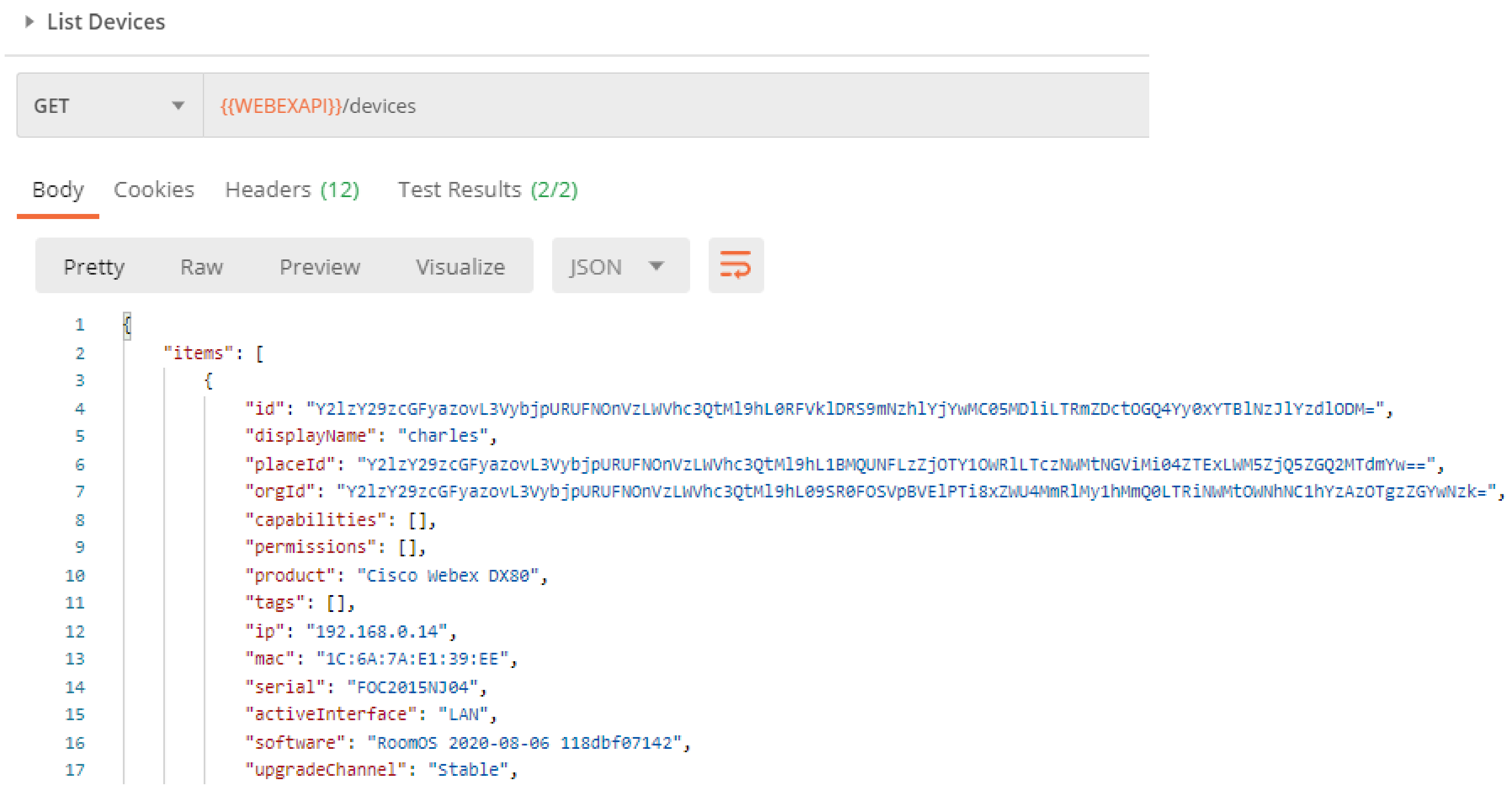
Imports: 50+

Webex Teams Cards

Give new levels of interactivity to Webex Teams with Cards and Actions

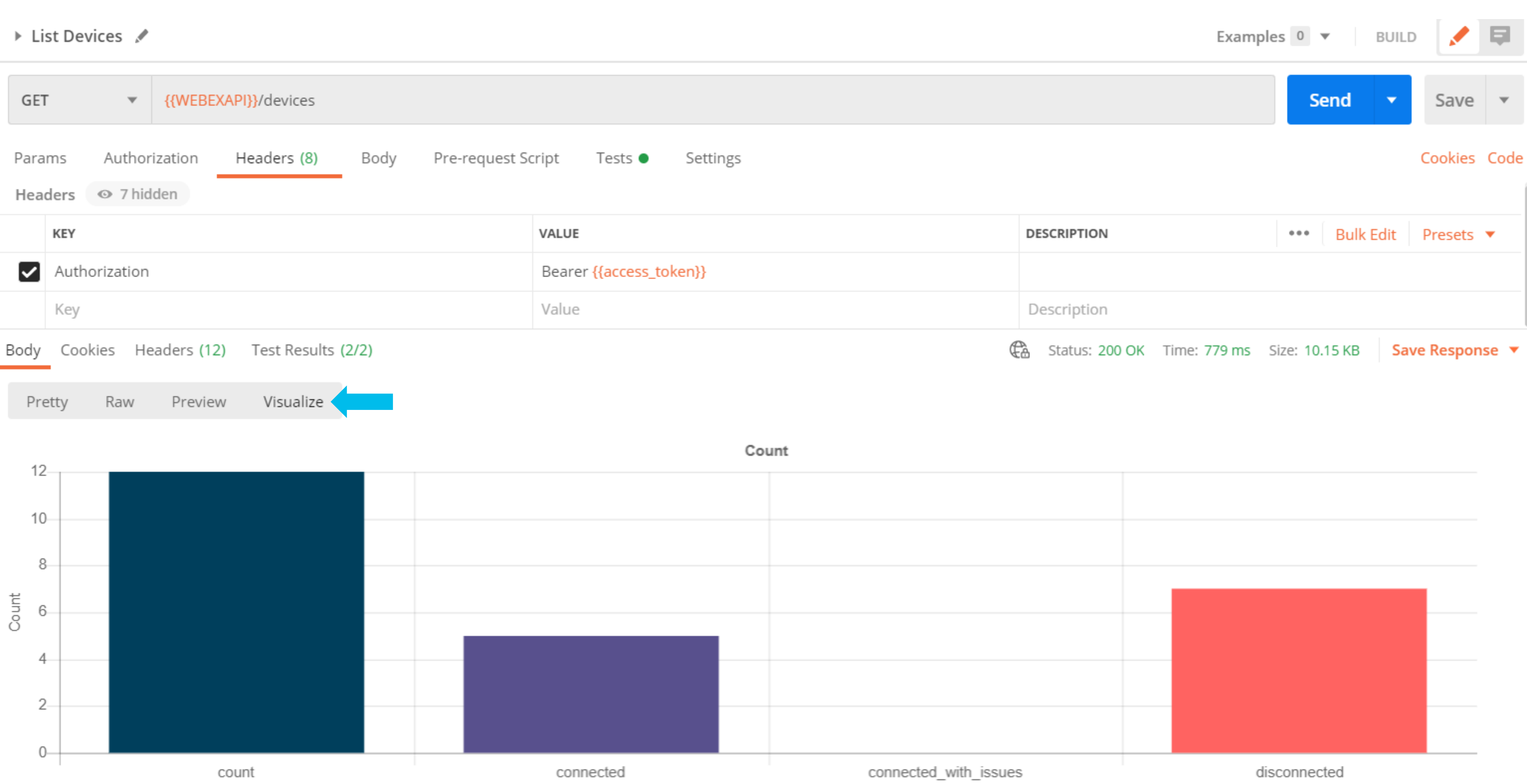
Imports: 40+

CHECK all collections for Cisco APIs:
https://explore.postman.com/team/ciscodevnet



{{WEBEXAPI}}/devices

```
{
    "items": [
        {
            "id": "Y2lzY29zcGFyazovL3VybjpURUFNOnVzLWVhc3QtMl9hL0RFVklDRS9mNzhlYjYwMC05MDliLTRmZDctOGQ4Yy0xYTBlNzJlYzdlODM=",
            "displayName": "charles",
            "placeId": "Y2lzY29zcGFyazovL3VybjpURUFNOnVzLWVhc3QtMl9hL1BMQUNFLzZjOTY1OWRlLTczNWMtNGViMi04ZTExLWM5ZjQ5ZGQ2MTdmYw==",
            "orgId": "Y2lzY29zcGFyazovL3VybjpURUFNOnVzLWVhc3QtMl9hL09SR0FOSVpBVElPTi8xZWU4MmRlMy1hMmQ0LTRiNWMtOWNhNC1hYzAzOTgzZGYwNzk=",
            "capabilities": [],
            "permissions": [],
            "product": "Cisco Webex DX80",
            "tags": [],
            "ip": "192.168.0.14",
            "mac": "1C:6A:7A:E1:39:EE",
            "serial": "FOC2015NJ04",
            "activeInterface": "LAN",
            "software": "RoomOS 2020-08-06 118dbf07142",
            "upgradeChannel": "Stable",
```

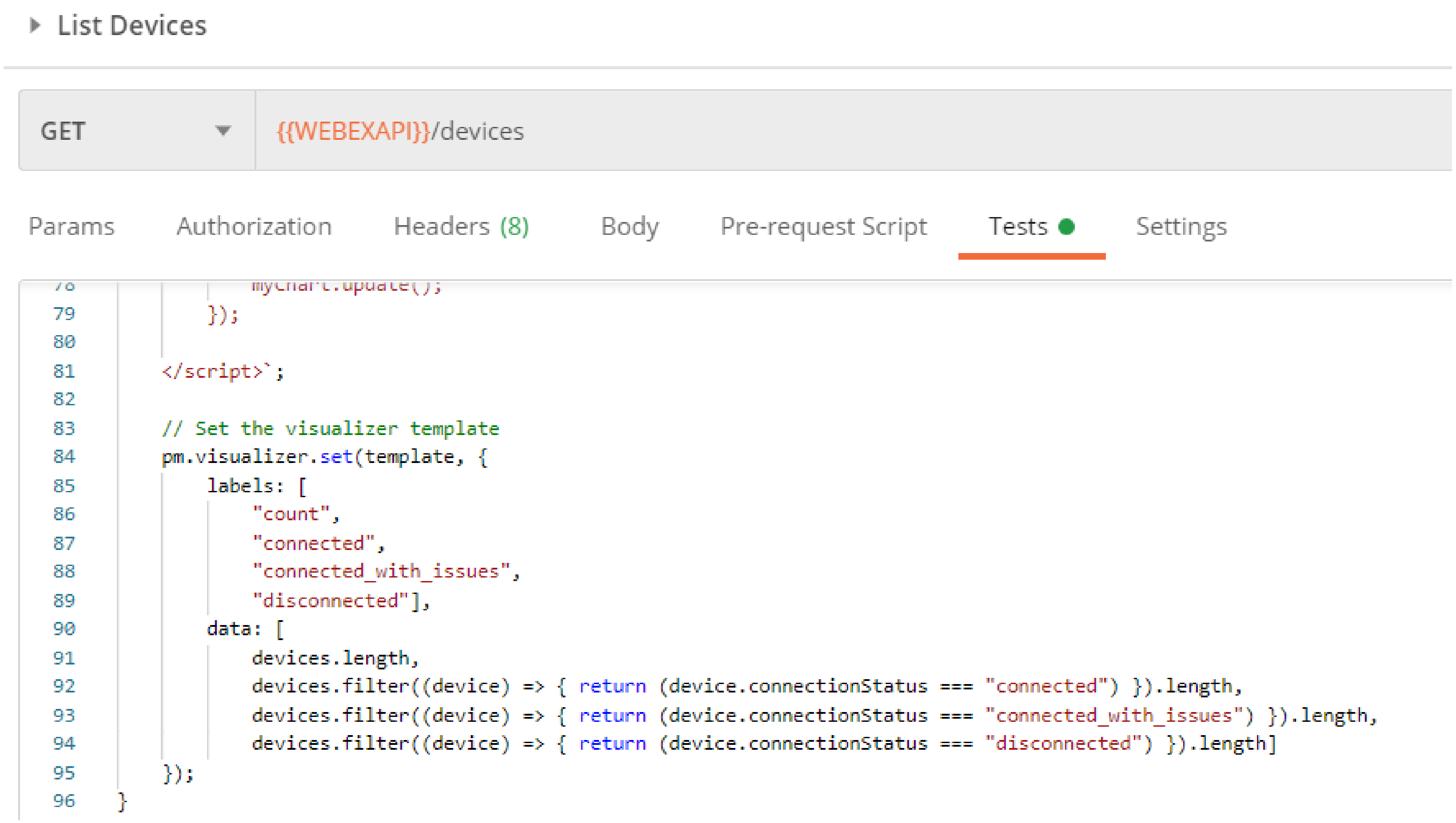


List Devices

GET {{WEBEXAPI}}/devices

	KEY	VALUE	DESCRIPTION
✓	Authorization	Bearer {{access_token}}	
	Key	Value	Description

Status: 200 OK Time: 779 ms Size: 10.15 KB



▸ List Devices

GET {{WEBEXAPI}}/devices

Params Authorization Headers (8) Body Pre-request Script Tests ● Settings

```
        myChart.update();
    });

</script>`;

// Set the visualizer template
pm.visualizer.set(template, {
    labels: [
        "count",
        "connected",
        "connected_with_issues",
        "disconnected"],
    data: [
        devices.length,
        devices.filter((device) => { return (device.connectionStatus === "connected") }).length,
        devices.filter((device) => { return (device.connectionStatus === "connected_with_issues") }).length,
        devices.filter((device) => { return (device.connectionStatus === "disconnected") }).length]
});
}
```

Cloud APIs for Webex Devices

Device Configurations API

released

/deviceConfigurations

GET

List Configurations for a device

PATCH

Update, Delete Configurations for a device

OAuth scopes

for Webex administrators:

- spark-admin:devices_read
- spark-admin:devices_write

```
curl -X PATCH
'/v1/deviceConfigurations?deviceId=Y2lzY…’
-H 'Authorization: Bearer YzViYz…’
-H 'Content-Type:  application/json-patch+json'
--data-raw '[{
 "op"     : "replace",
 "path"   : "Audio.Ultrasound.MaxVolume/sources
            /configured/value",
 "value" : 70 }]'
```

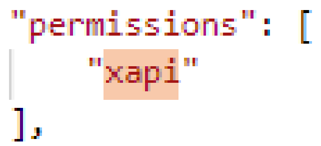
Cloud /xAPI for Webex Devices

https://developer.webex.com/docs/api/guides/xapi

/xapi

GET
Query Status

POST
Execute Command



```
"permissions": [
    "xapi"
],
```

<u>OAuth scopes</u>

- spark:xapi_statuses
- spark:xapi_commands

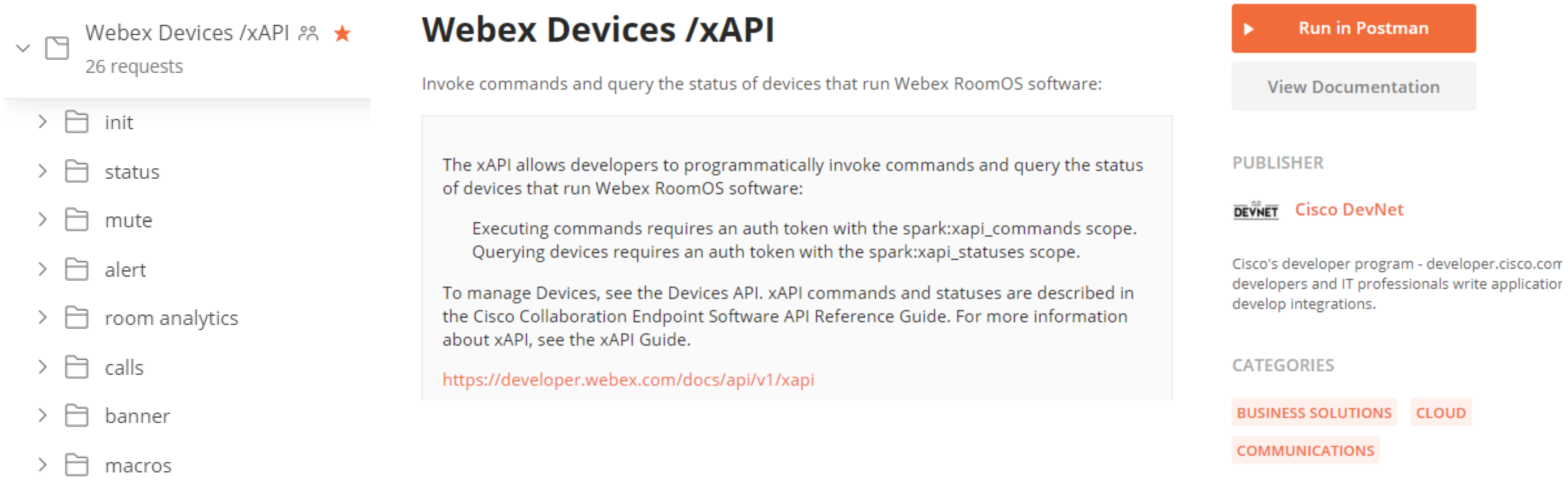
Webex administrators can invoke GET & POST /xapi for ALL devices listed via GET /devices

- no need for the 'xapi' to show up for the permission attribute

Webex users & bots can invoke GET or POST /xapi

- for devices listed with GET /devices
- with 'xapi' displayed for the permission attribute (manually configured in Webex Control Hub)

Postman collections for Webex /xAPI


Webex Devices /xAPI
26 requests

- init
- status
- mute
- alert
- room analytics
- calls
- banner
- macros
- httpclient
- message

Webex Devices /xAPI

Invoke commands and query the status of devices that run Webex RoomOS software:

The xAPI allows developers to programmatically invoke commands and query the status of devices that run Webex RoomOS software:

> Executing commands requires an auth token with the spark:xapi_commands scope. Querying devices requires an auth token with the spark:xapi_statuses scope.

To manage Devices, see the Devices API. xAPI commands and statuses are described in the Cisco Collaboration Endpoint Software API Reference Guide. For more information about xAPI, see the xAPI Guide.

https://developer.webex.com/docs/api/v1/xapi

PUBLISHER

Cisco DevNet

Cisco's developer program - developer.cisco.com
developers and IT professionals write applicatior
develop integrations.

CATEGORIES

BUSINESS SOLUTIONS CLOUD COMMUNICATIONS

CHECK all collections for Cisco APIs
https://explore.postman.com/team/ciscodevnet

Query Status
Execute Commands
[demo]

Webex APIs Access Tokens

Detailed in DGTL-DEVNET-2610

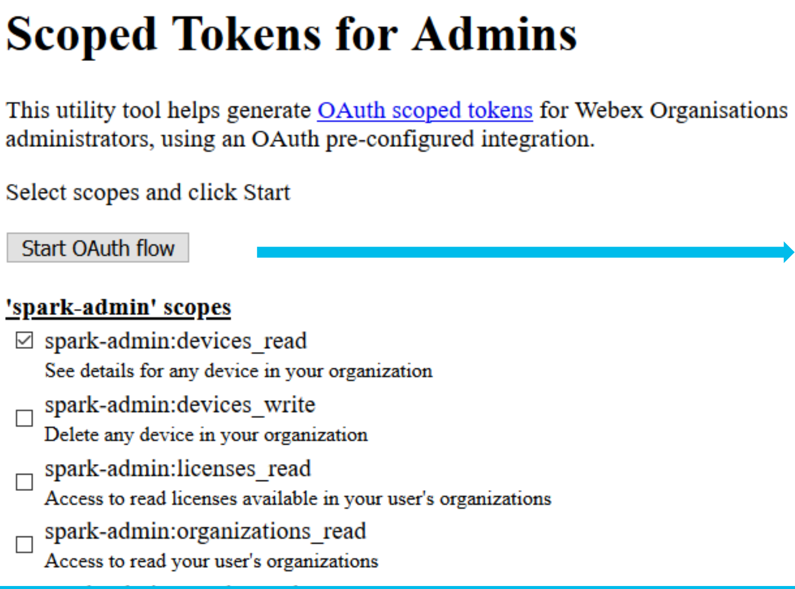
Testing	Thanks to your developer token, invoke Webex APIs **under your own Webex identity.**
Integration	Invoke Webex APIs **on behalf of Webex Users' identity.** Supports fine-grained permissions **using OAuth scopes.**
Bot Account	Invoke the Webex APIs under a **machine identity. Inherits only the 'user-level' OAuth scopes** of its creator (admin privileges are NOT transferred).
Guest Account	Allows for **non-Webex users** to use the Messaging, Calling and Meeting services (note that at least one Webex user must be taking part in the activity).

webex-integration-admin

Utility to create 'scoped' OAuth tokens for Webex Admins and Compliance Officers

Detailed in DGTL-DEVNET-2610

https://developer.cisco.com/codeexchange



Scoped Tokens for Admins

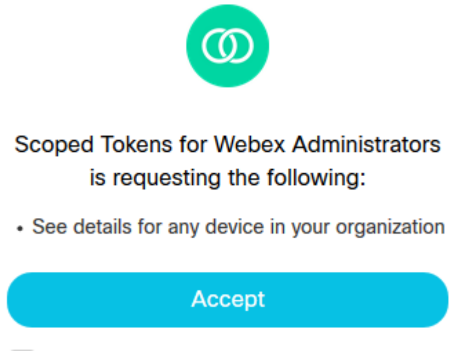
This utility tool helps generate [OAuth scoped tokens] for Webex Organisations administrators, using an OAuth pre-configured integration.

Select scopes and click Start

Start OAuth flow

'spark-admin' scopes

- ☑ spark-admin:devices_read
  See details for any device in your organization
- ☐ spark-admin:devices_write
  Delete any device in your organization
- ☐ spark-admin:licenses_read
  Access to read licenses available in your user's organizations
- ☐ spark-admin:organizations_read
  Access to read your user's organizations



Scoped Tokens for Webex Administrators is requesting the following:

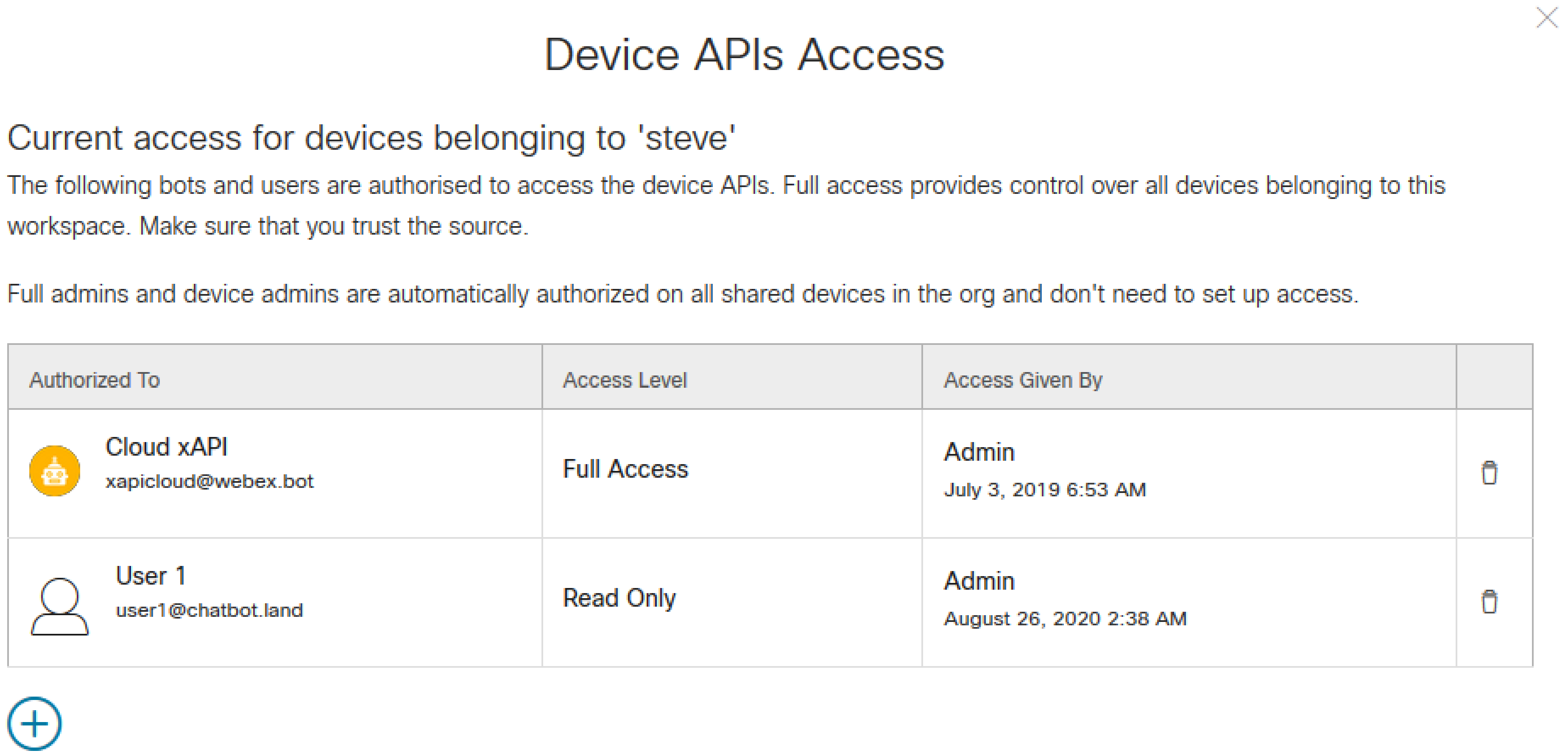
- See details for any device in your organization

Accept

https://webex-token.herokuapp.com

Webex APIs Access Tokens

Testing	Thanks to your developer token, invoke Webex APIs **under your own Webex identity.**
Integration	Invoke Webex APIs **on behalf of Webex Users' identity.** Supports fine-grained permissions **using OAuth scopes.**
Bot Account	Invoke the Webex APIs under a **machine identity. Inherits only the 'user-level' OAuth scopes** of its creator (admin privileges are NOT transferred).
Guest Account	Allows for **non-Webex users** to use the Messaging, Calling and Meeting services (note that at least one Webex user must be taking part in the activity).



Device APIs Access

Current access for devices belonging to 'steve'

The following bots and users are authorised to access the device APIs. Full access provides control over all devices belonging to this workspace. Make sure that you trust the source.

Full admins and device admins are automatically authorized on all shared devices in the org and don't need to set up access.

Authorized To	Access Level	Access Given By	
Cloud xAPI xapicloud@webex.bot	Full Access	Admin July 3, 2019 6:53 AM	
User 1 user1@chatbot.land	Read Only	Admin August 26, 2020 2:38 AM	

Support for multi-line commands

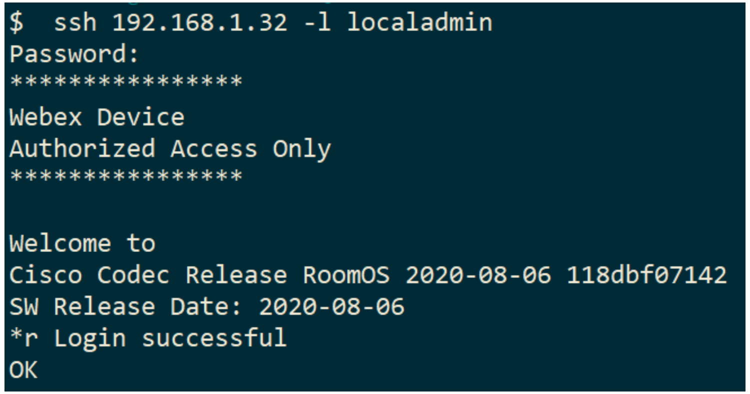
new

POST /v1/xapi/command/SystemUnit.WelcomeBanner.Set

```
{
    "deviceId": " Y2lzY29zcGF…yazoQzNWU=",
    "arguments": {},
    "body": "***************\nWebex Device\nAuthorized Access Only\n***************"
}
```

200 OK

```
{
    "deviceId": "Y2lzY29zcGF…yazoQzNWU=",
    "result": {}
}
```



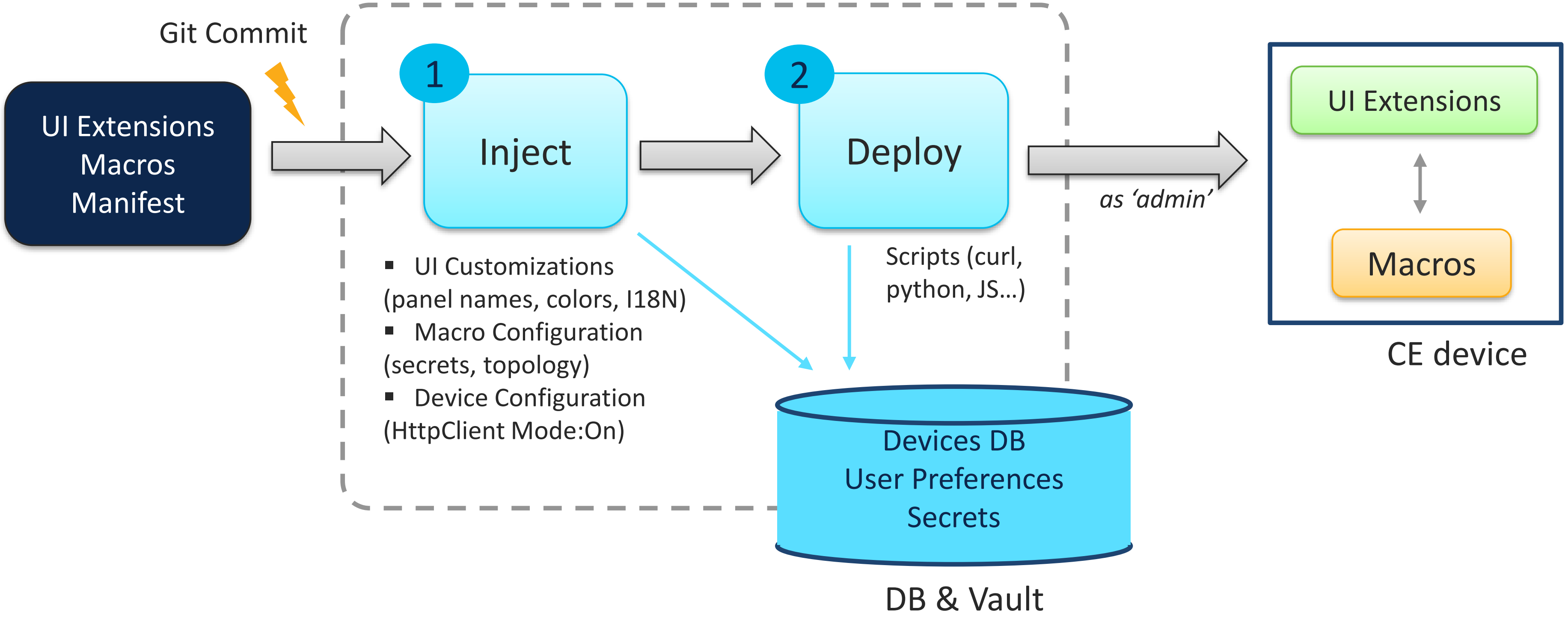
```
$  ssh 192.168.1.32 -l localadmin
Password:
****************
Webex Device
Authorized Access Only
****************

Welcome to
Cisco Codec Release RoomOS 2020-08-06 118dbf07142
SW Release Date: 2020-08-06
*r Login successful
OK
```

[Use case] Deploying UI Extensions & Macros over the cloud

Deployment strategies

Inject topology and secrets along the CI/CD pipeline

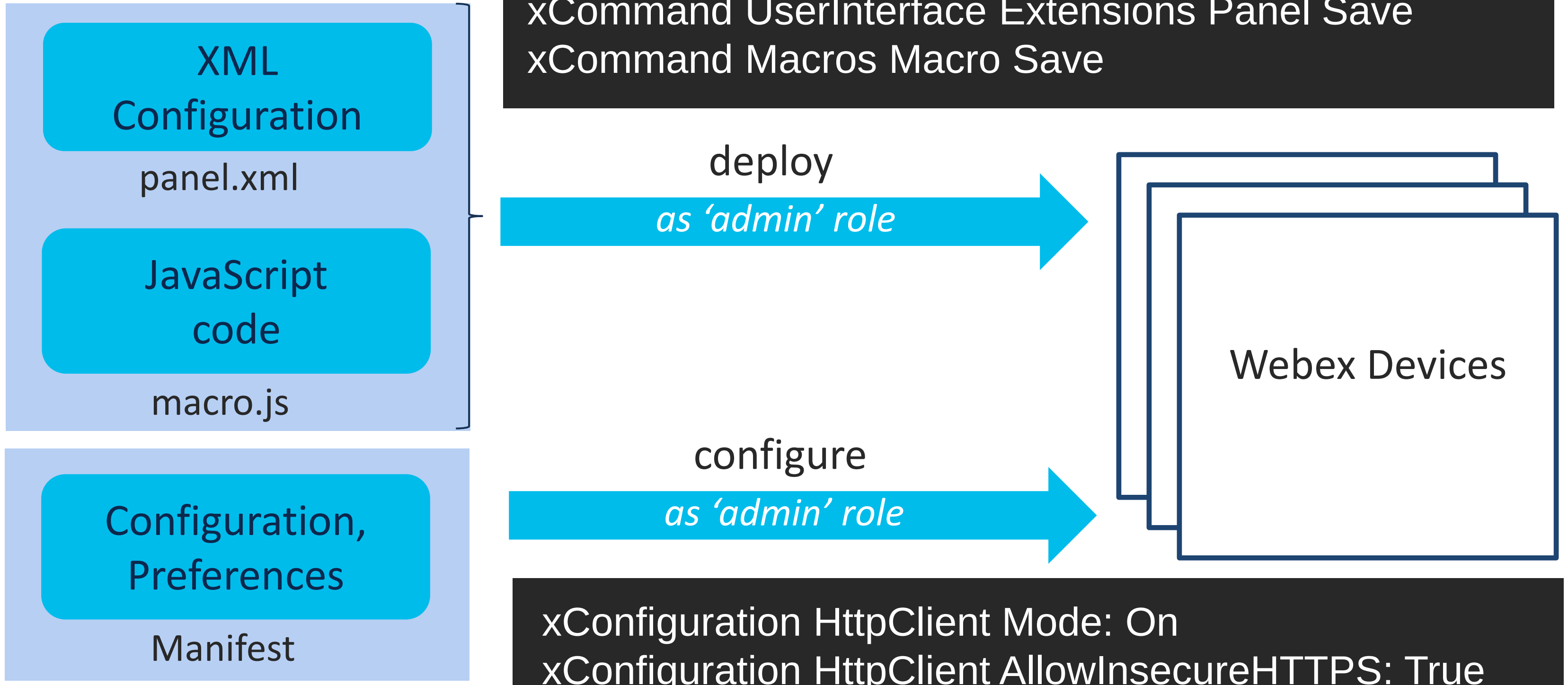
See BRKPRG-3244
CL20B



- UI Customizations (panel names, colors, I18N)
- Macro Configuration (secrets, topology)
- Device Configuration (HttpClient Mode:On)

Scripts (curl, python, JS…)

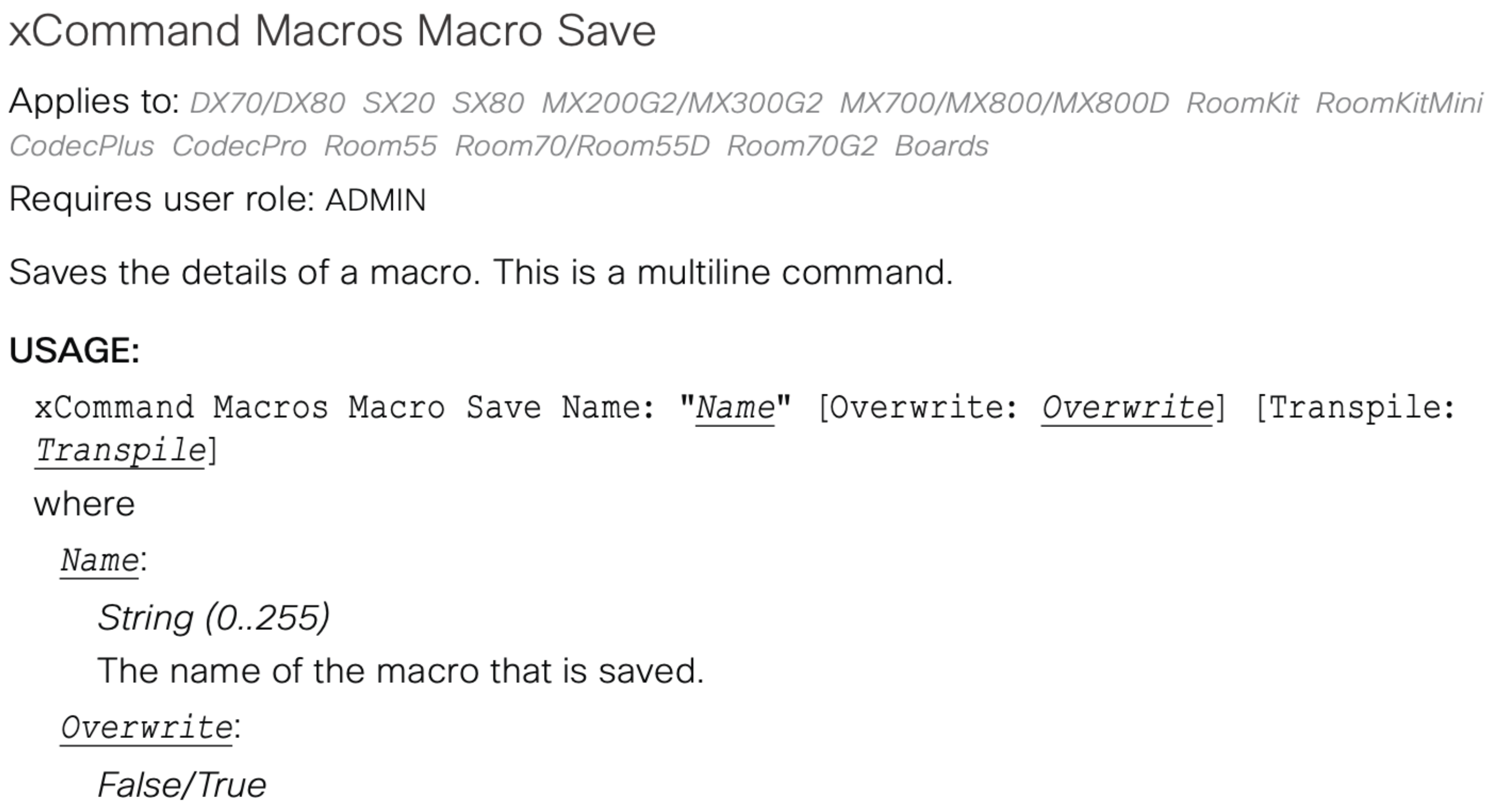
Deploying to Webex Devices



```
xCommand UserInterface Extensions Panel Save
xCommand Macros Macro Save
```

```
xConfiguration HttpClient Mode: On
xConfiguration HttpClient AllowInsecureHTTPS: True
```

Automated deployment of Macros



xCommand Macros Macro Save

Applies to: *DX70/DX80 SX20 SX80 MX200G2/MX300G2 MX700/MX800/MX800D RoomKit RoomKitMini CodecPlus CodecPro Room55 Room70/Room55D Room70G2 Boards*

Requires user role: ADMIN

Saves the details of a macro. This is a multiline command.

USAGE:

```
xCommand Macros Macro Save Name: "Name" [Overwrite: Overwrite] [Transpile:
Transpile]
```

where

Name:

String (0..255)

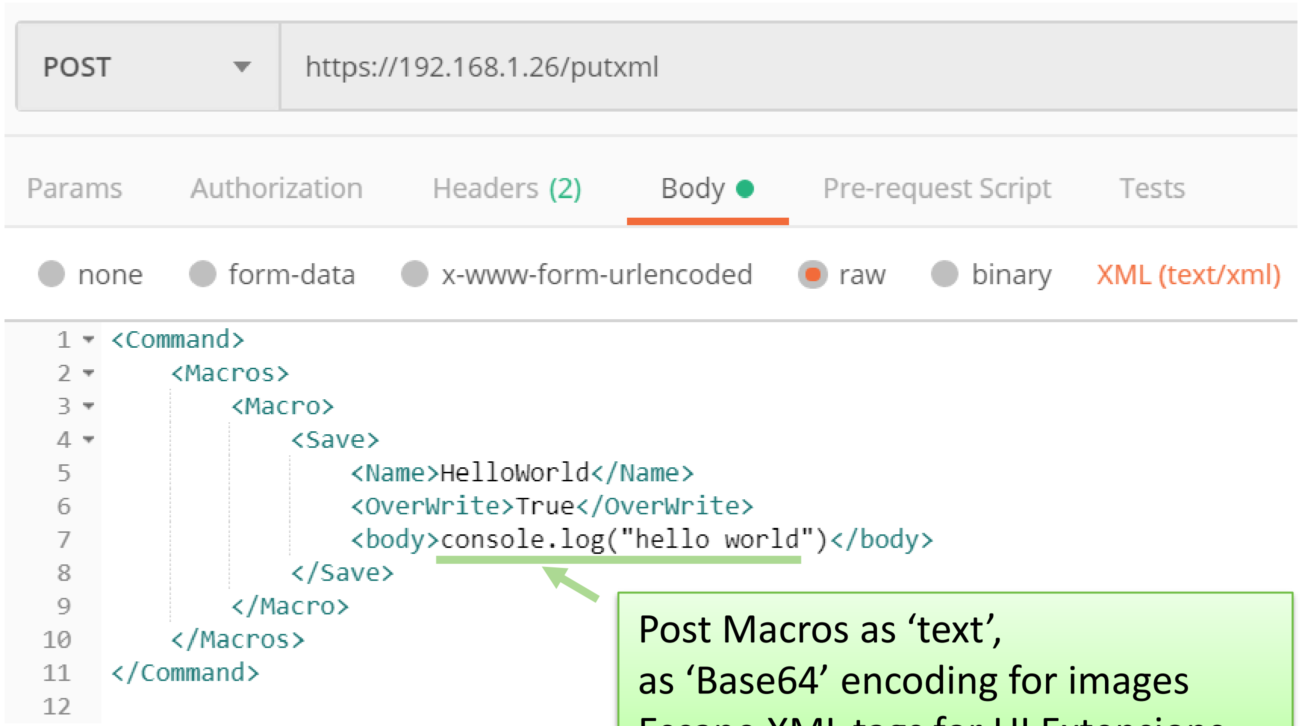
The name of the macro that is saved.

Overwrite:

False/True

XML over HTTP: /putxml

LAN Access



```
<Command>
    <Macros>
        <Macro>
            <Save>
                <Name>HelloWorld</Name>
                <OverWrite>True</OverWrite>
                <body>console.log("hello world")</body>
            </Save>
        </Macro>
    </Macros>
</Command>
```

Post Macros as 'text',
as 'Base64' encoding for images
Escape XML tags for UI Extensions.

serial port
ssh
HTTP
WebSocket

text
XML
JSON

Support for multi-line commands

Webex Devices /xAPI

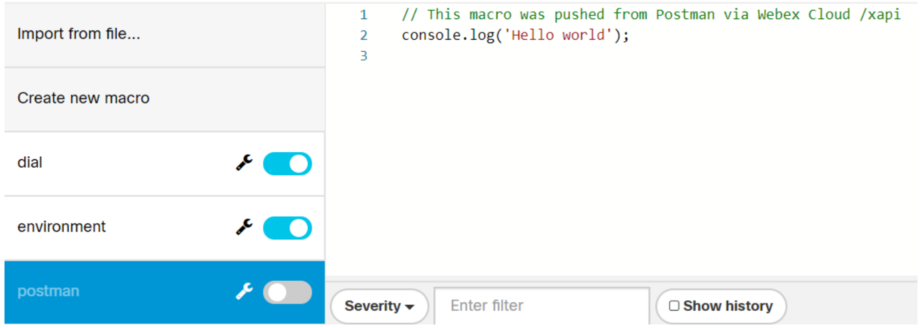
new

POST /v1/xapi/command/Macros.Macro.Save

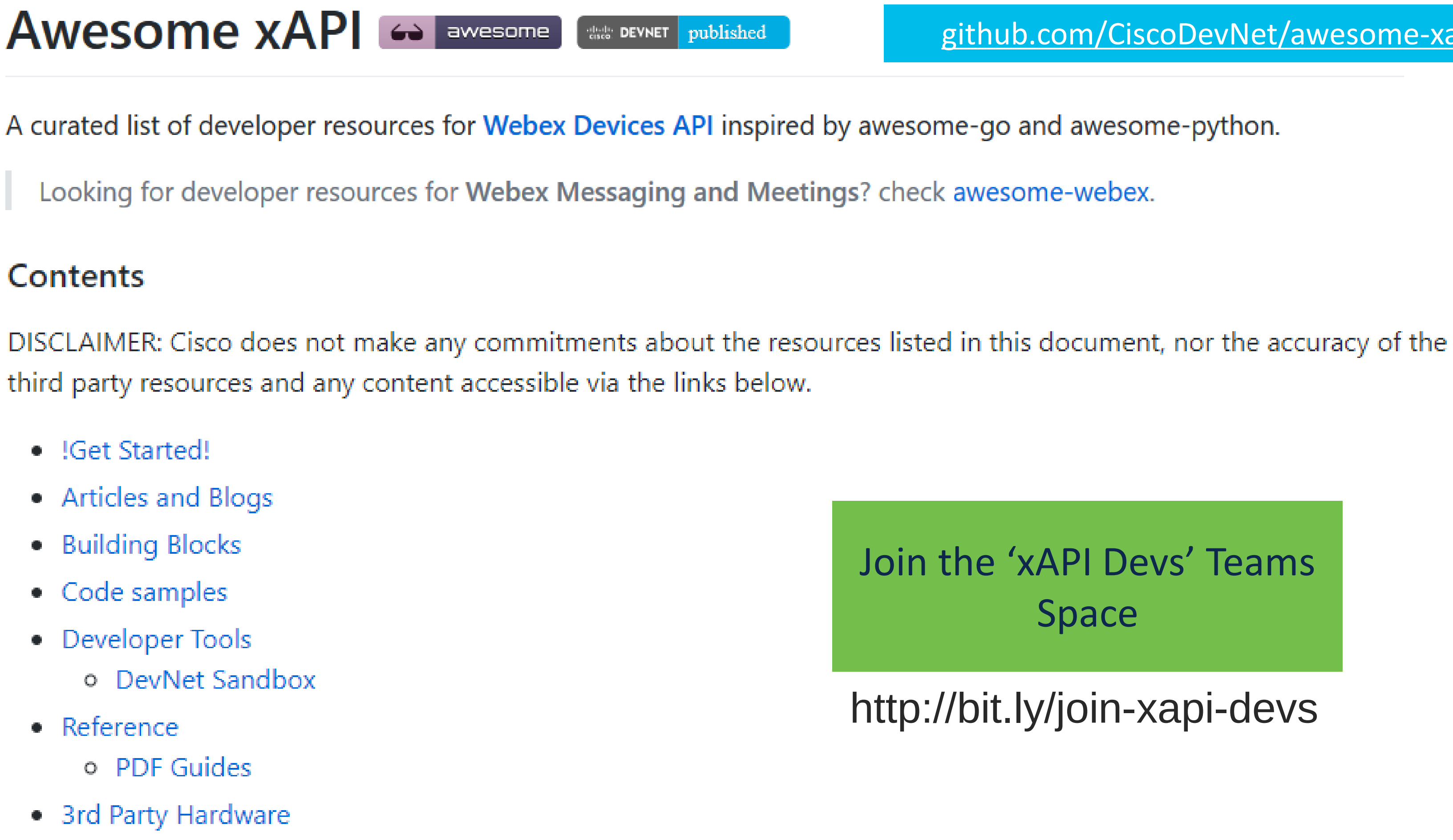
```
{
    "deviceId": "Y2lzY29zcGF…yazoQzNWU=",
    "arguments": {
        "Name": "postman",
        "OverWrite": "False"
    },
    "body": "// This macro was pushed from P
ostman via Webex Cloud /xapi\nconsole.log('H
ello world');\n"
}
```

200 OK

```
{
    "deviceId": "Y2lzY29zcGF…yazoQzNWU=",
    "result": {}
}
```



```
// This macro was pushed from Postman via Webex Cloud /xapi
console.log('Hello world');
```



Awesome xAPI

A curated list of developer resources for Webex Devices API inspired by awesome-go and awesome-python.

> Looking for developer resources for **Webex Messaging and Meetings**? check awesome-webex.

Contents

DISCLAIMER: Cisco does not make any commitments about the resources listed in this document, nor the accuracy of the third party resources and any content accessible via the links below.

- !Get Started!
- Articles and Blogs
- Building Blocks
- Code samples
- Developer Tools
  - DevNet Sandbox
- Reference
  - PDF Guides
- 3rd Party Hardware

github.com/CiscoDevNet/awesome-xapi

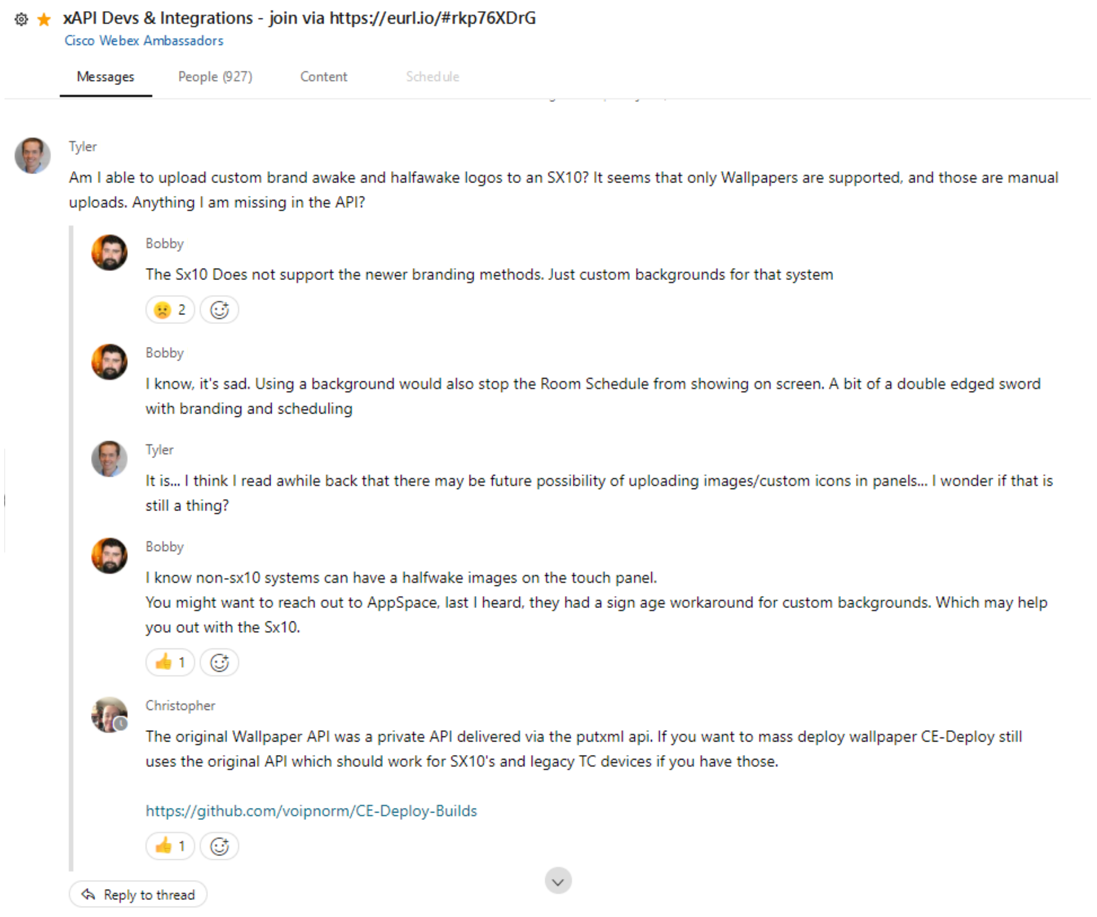
Join the 'xAPI Devs' Teams Space

http://bit.ly/join-xapi-devs

‘xAPI Devs’

- Join the community
  https://eurl.io/#rkp76XDrG
  http://bit.ly/join-xapi-devs

- 900+ members



xAPI Devs & Integrations - join via https://eurl.io/#rkp76XDrG
Cisco Webex Ambassadors

Tyler
Am I able to upload custom brand awake and halfawake logos to an SX10? It seems that only Wallpapers are supported, and those are manual uploads. Anything I am missing in the API?

Bobby
The Sx10 Does not support the newer branding methods. Just custom backgrounds for that system

😟 2

Bobby
I know, it's sad. Using a background would also stop the Room Schedule from showing on screen. A bit of a double edged sword with branding and scheduling

Tyler
It is... I think I read awhile back that there may be future possibility of uploading images/custom icons in panels... I wonder if that is still a thing?

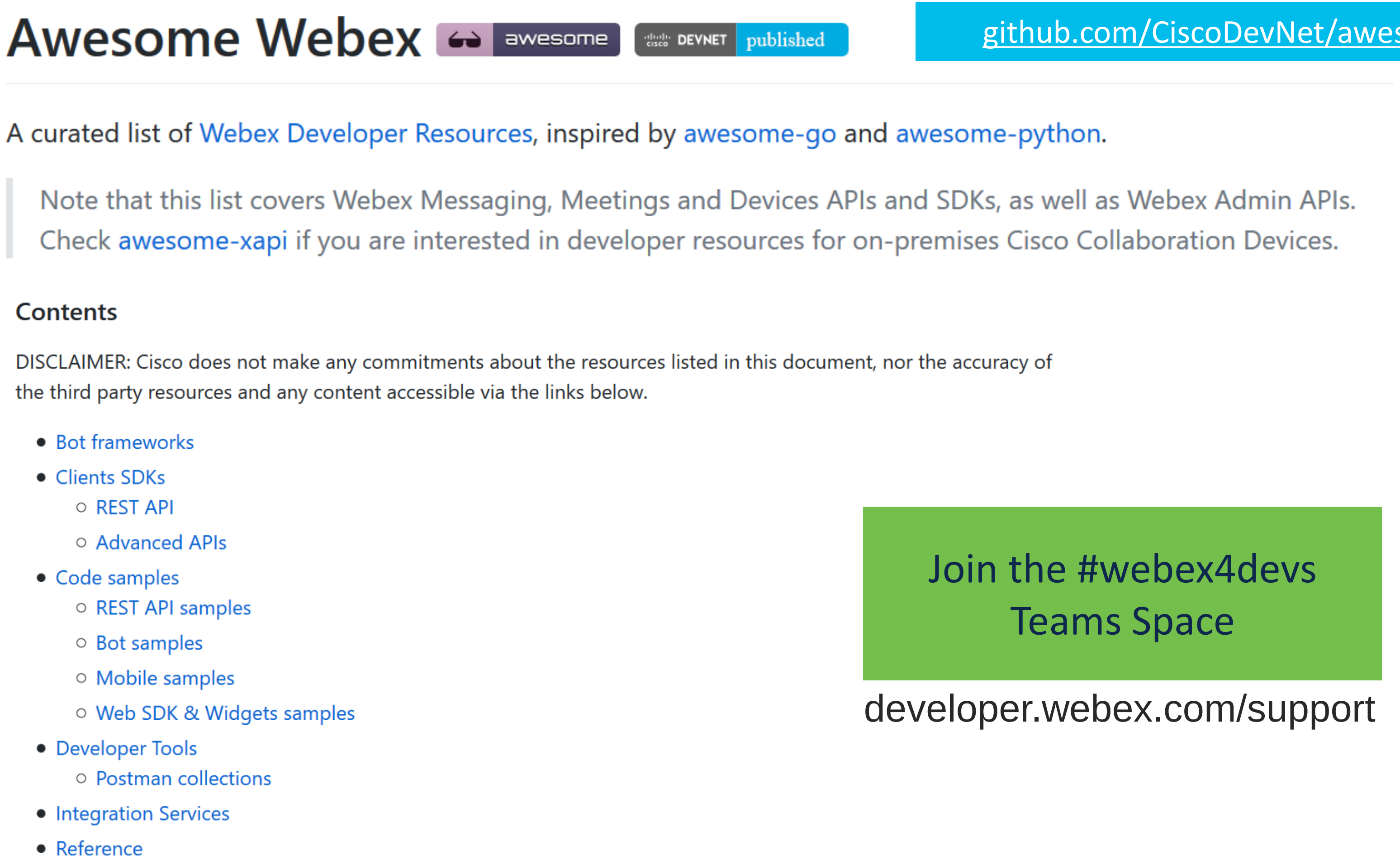
Bobby
I know non-sx10 systems can have a halfwake images on the touch panel.
You might want to reach out to AppSpace, last I heard, they had a sign age workaround for custom backgrounds. Which may help you out with the Sx10.

👍 1

Christopher
The original Wallpaper API was a private API delivered via the putxml api. If you want to mass deploy wallpaper CE-Deploy still uses the original API which should work for SX10's and legacy TC devices if you have those.

https://github.com/voipnorm/CE-Deploy-Builds

👍 1



Awesome Webex

A curated list of Webex Developer Resources, inspired by awesome-go and awesome-python.

> Note that this list covers Webex Messaging, Meetings and Devices APIs and SDKs, as well as Webex Admin APIs. Check awesome-xapi if you are interested in developer resources for on-premises Cisco Collaboration Devices.

Contents

DISCLAIMER: Cisco does not make any commitments about the resources listed in this document, nor the accuracy of the third party resources and any content accessible via the links below.

github.com/CiscoDevNet/awesome-webex

Join the #webex4devs Teams Space

developer.webex.com/support

Thank you

CISCO

Possibilities

#CiscoLive