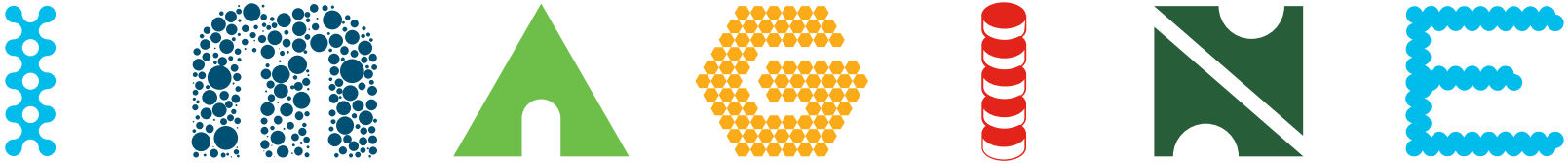
Cisco live!

January 28 - February 1, 2019 - Barcelona

cisco



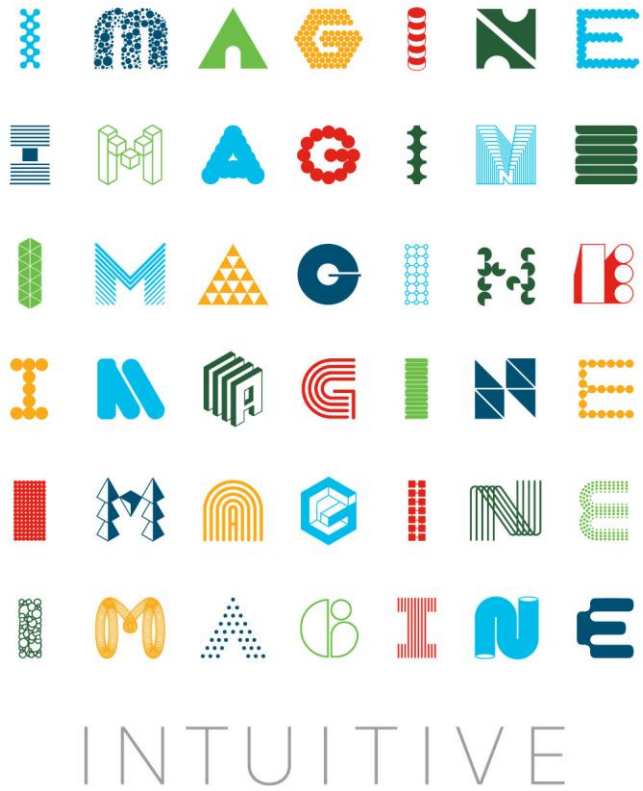
INTUITIVE

DEVWKS-2074

Enhancing Meeting Rooms User Experience with xAPI and Macros

Workshop

Stève Sfartz - Developer Experience Manager
& API Design Lead at Cisco DevNet



/Cisco/DevNet/SteveSfartz

- Manager of a WW team of Developer Advocates
  - IoT, Collab, Cloud, DevOps
  - Application Developer Use Cases
- Technical lead for the Cisco « API Style Guide »
- API veteran, hobbyist coder, hands-On Architect
- Contributor to DevNet CodeExchange
  - Code samples, developer tools, postman collections, awesome-webex, awesome-xapi…
- Based in France, DevNet Europe point of contact

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

"vision without execution is hallucination"

-- Thomas Edison

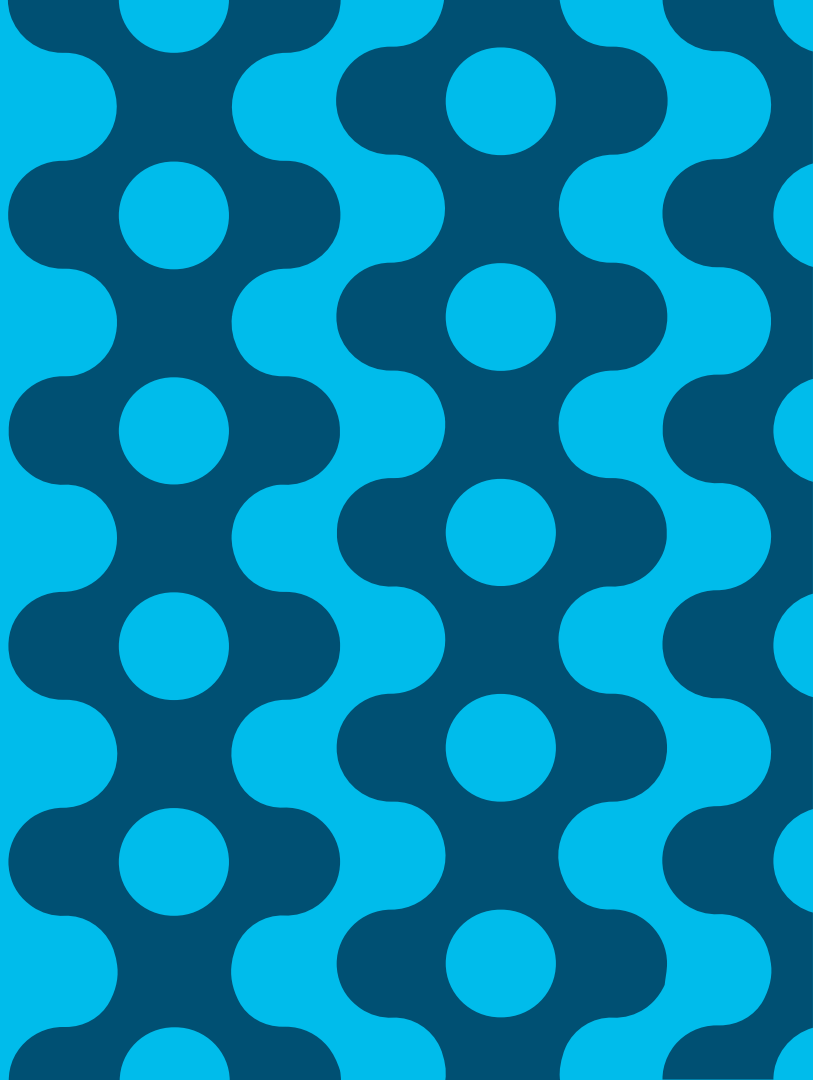
Agenda

- Meet your Sandbox'ed RoomKit
- Build a custom In-Room Control
- Listen to events via SSH
- Create your first Macro
- Post to a Webex Teams space with HttpClient

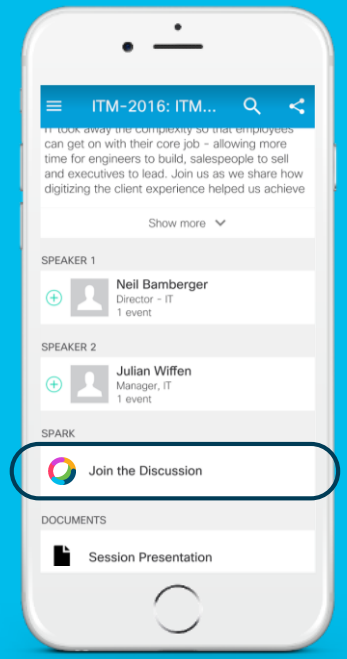
DEVWKS-2074

Continuing Education Credits for CCIEs

Please find an Ambassador to scan you in to this DEVWKS workshop to ensure you earn your Continuing Education Credit



Cisco live!



cs.co/ciscolivebot#DEVWKS-2074

Cisco Webex Teams

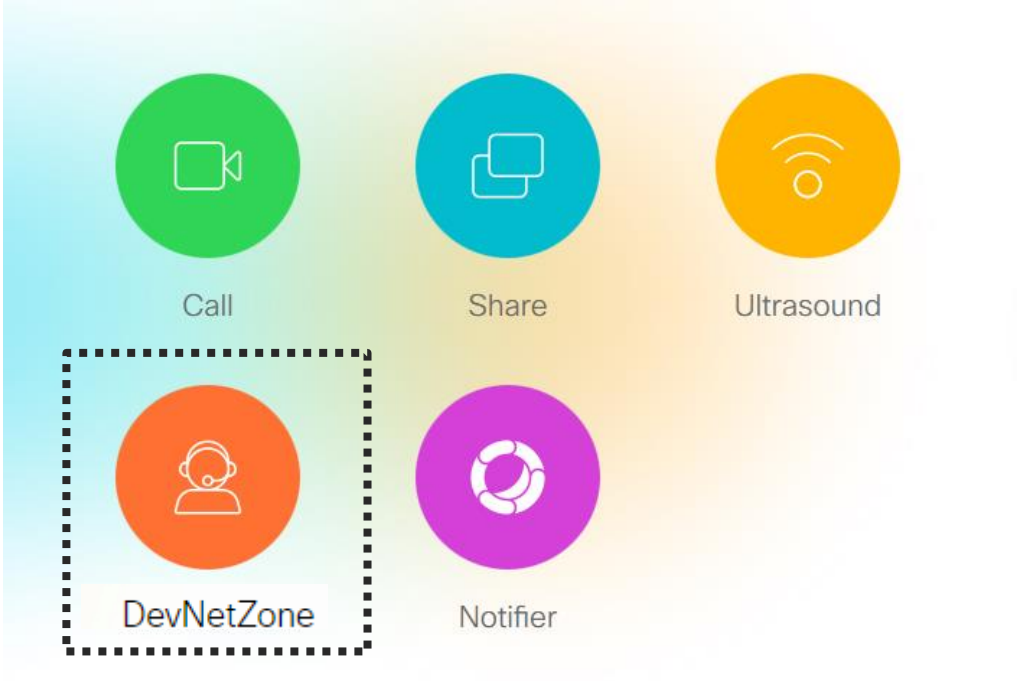
Questions?

Use Cisco Webex Teams (formerly Cisco Spark) to chat with the speaker after the session

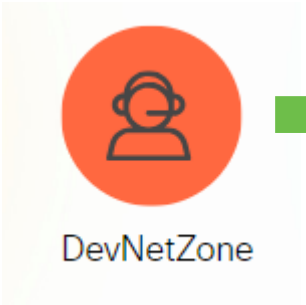
How

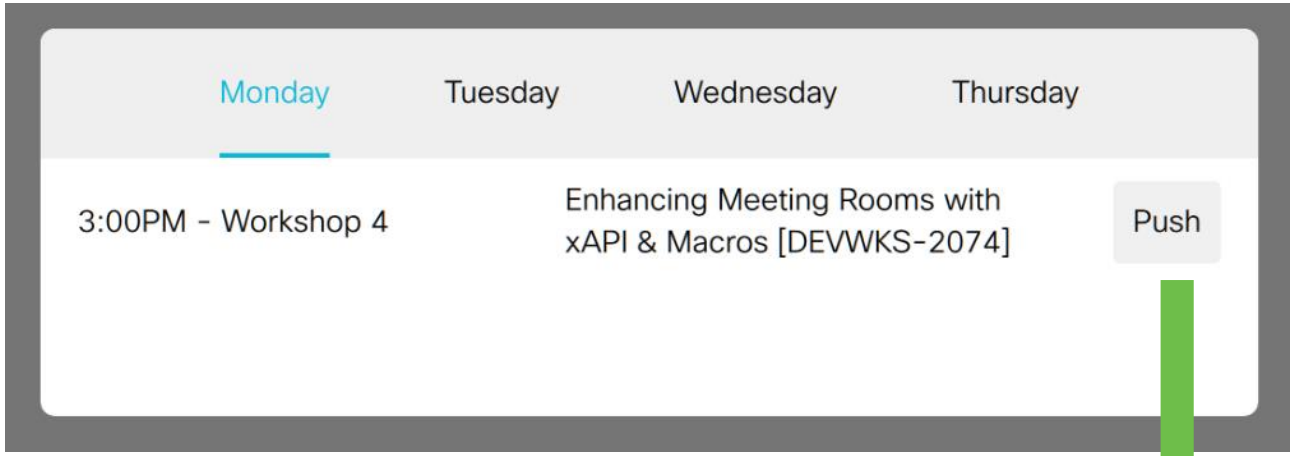
1. Find this session in the Cisco Events Mobile App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space

Our goal for today



Our goal for today





Our goal for today

POST /messages to a Webex Teams space

DevNetZoneLab 17:38

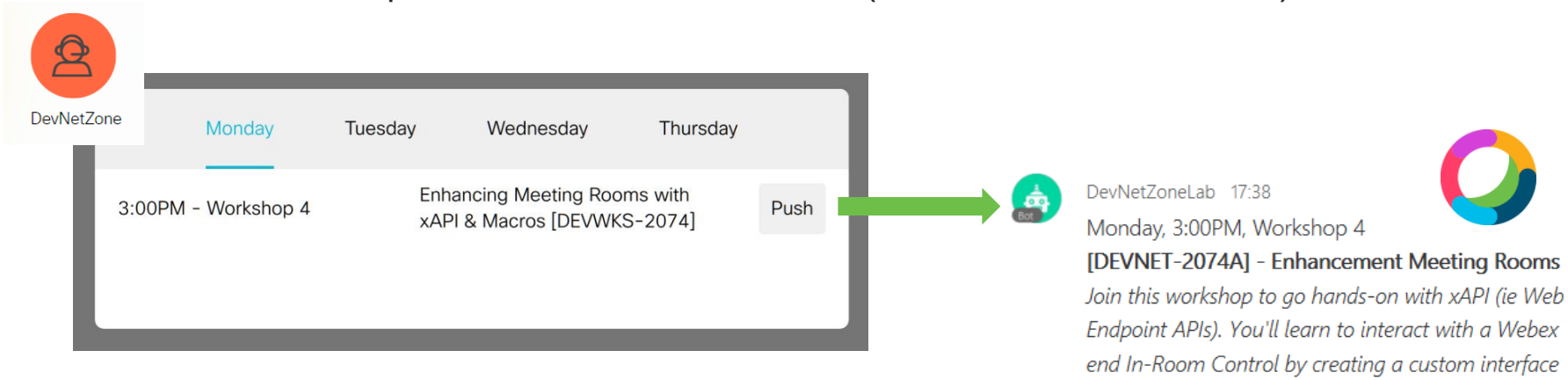
Monday, 3:00PM, Workshop 4

[DEVNET-2074A] - Enhancement Meeting Rooms User Experience with xAPI and Macros

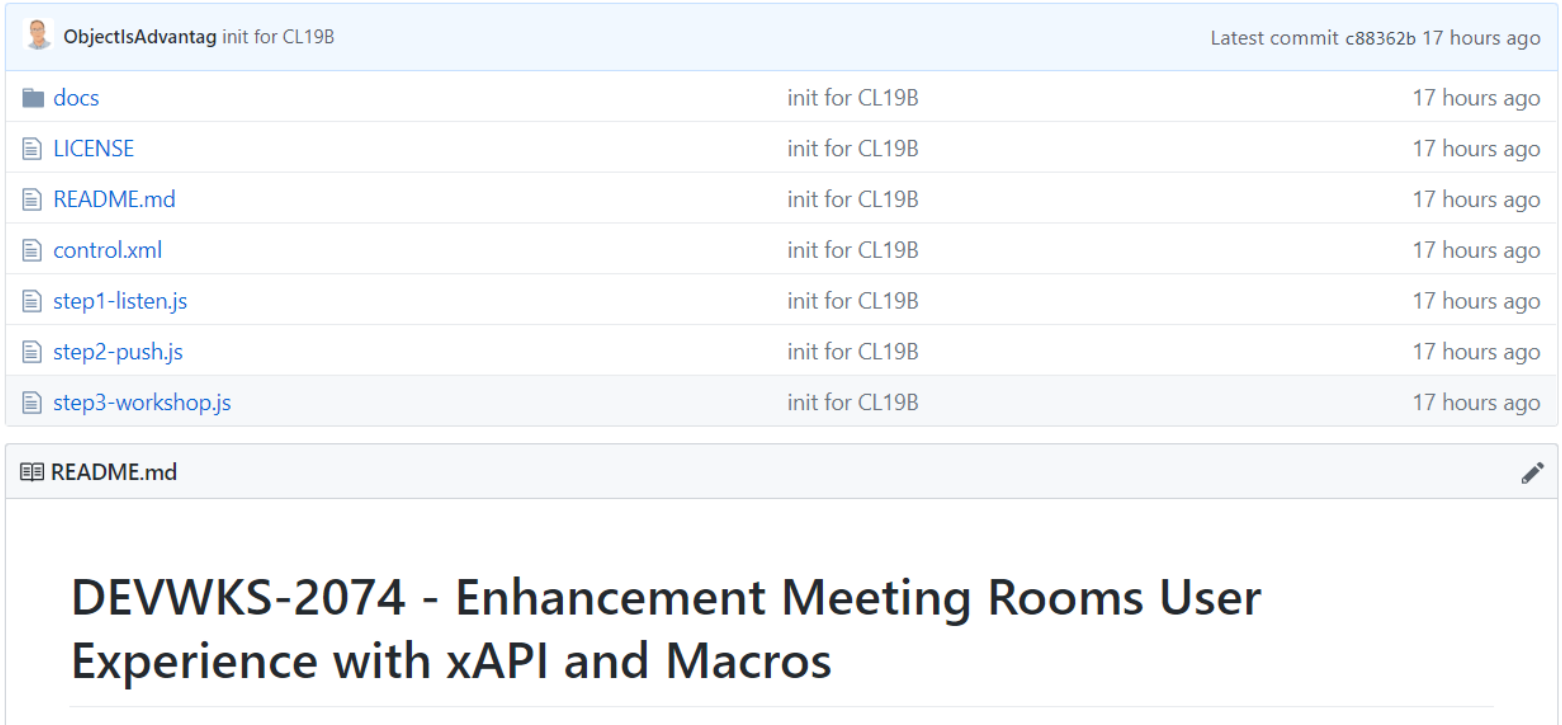
Join this workshop to go hands-on with xAPI (ie Webex Devices APIs and Cisco Collaboration Endpoint APIs). You'll learn to interact with a Webex Device from code, and implement an end-to-end In-Room Control by creating a custom interface and deploying Macros onto your device

Our goal for today

- Connect to a CE 9.6 RoomKit from the DevNet Sandbox
- Build and deploy a custom In-Room Control (CE 9.2.1+)
- Create and run Javascript Macros (CE 9.2.1+)
- Learn the new HttpClient POST command (comes with CE 9.6.1)



https://github.com/ObjectIsAdvantag/devnetzone-workshop-2074



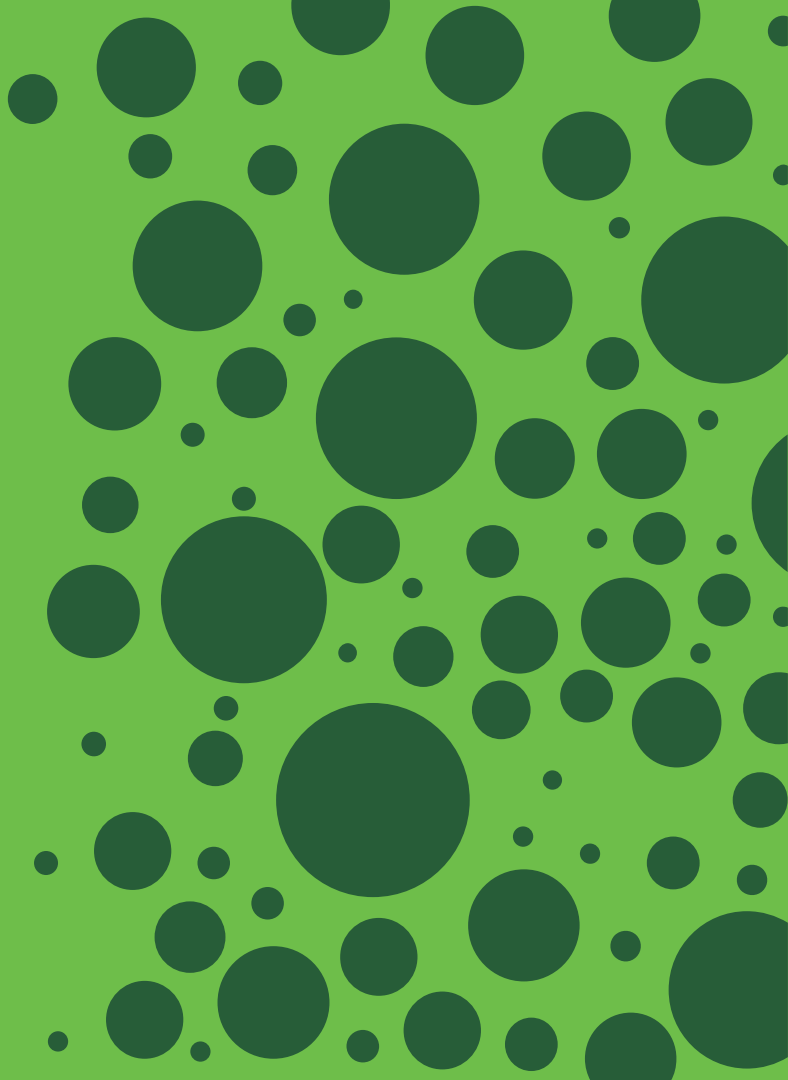
ObjectIsAdvantag init for CL19B — Latest commit c88362b 17 hours ago

docs	init for CL19B	17 hours ago
LICENSE	init for CL19B	17 hours ago
README.md	init for CL19B	17 hours ago
control.xml	init for CL19B	17 hours ago
step1-listen.js	init for CL19B	17 hours ago
step2-push.js	init for CL19B	17 hours ago
step3-workshop.js	init for CL19B	17 hours ago

README.md

DEVWKS-2074 - Enhancement Meeting Rooms User Experience with xAPI and Macros

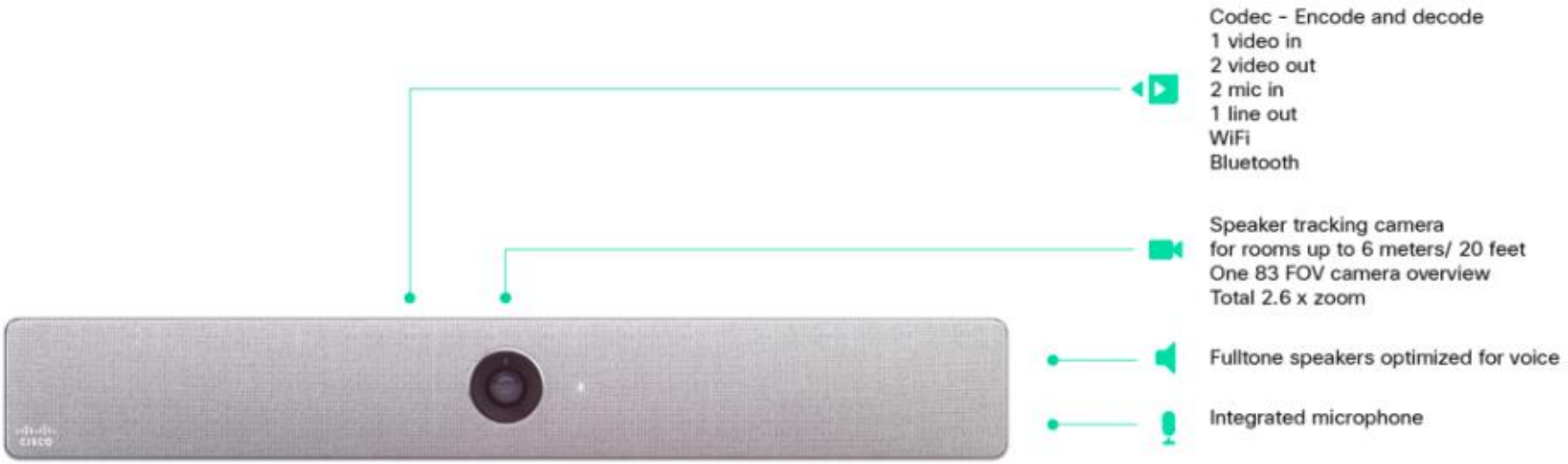
Meet your Cisco Collaboration Device




https://projectworkplace.cisco.com

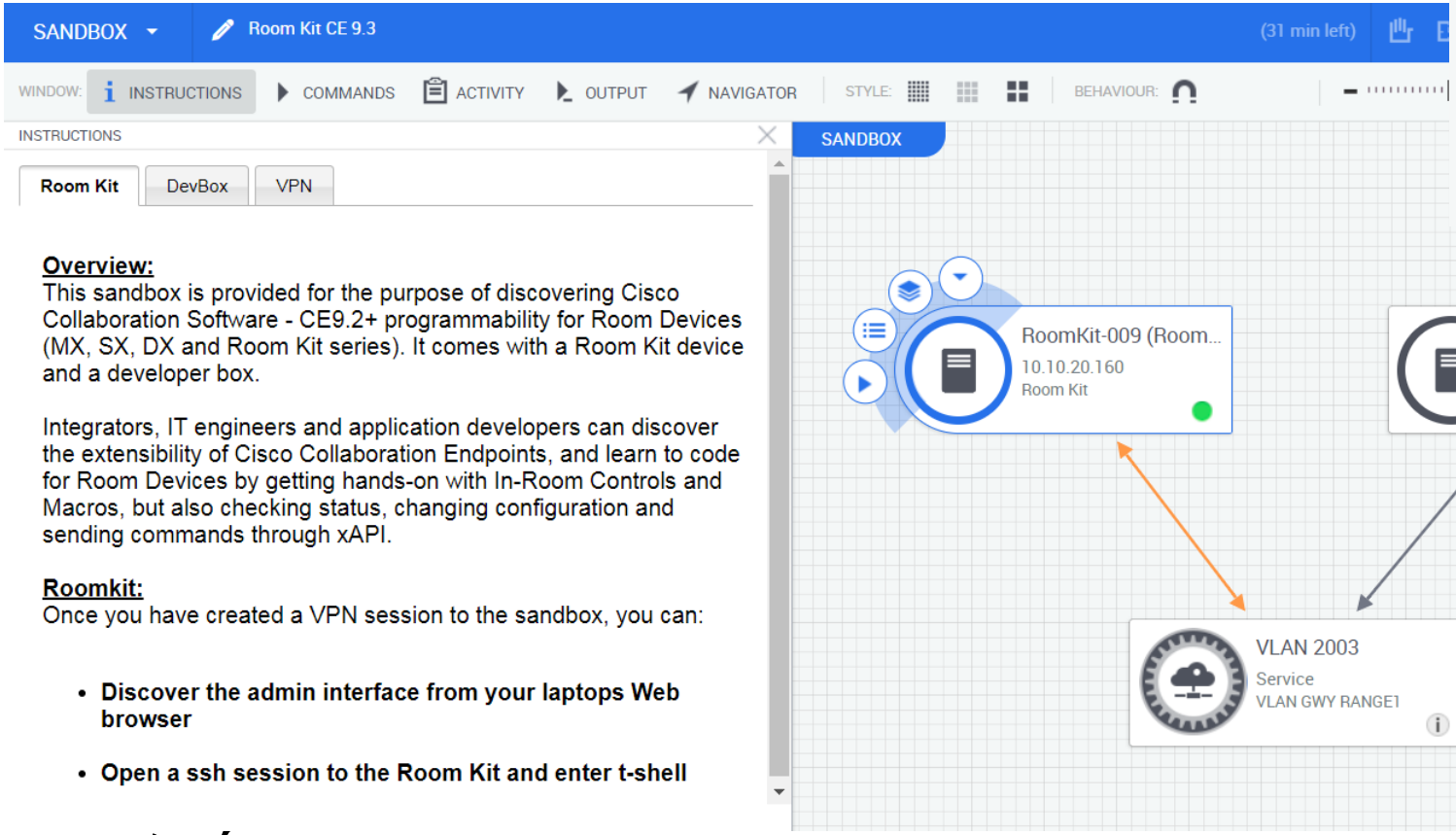
Webex Room Kit

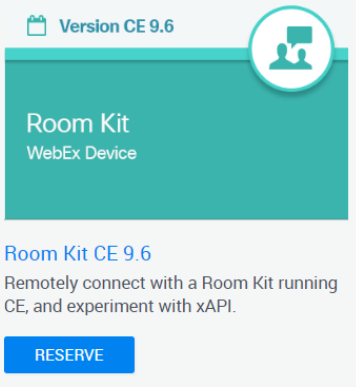
https://projectworkplace.cisco.com/#/en-us/product/webexroomkit/0/0



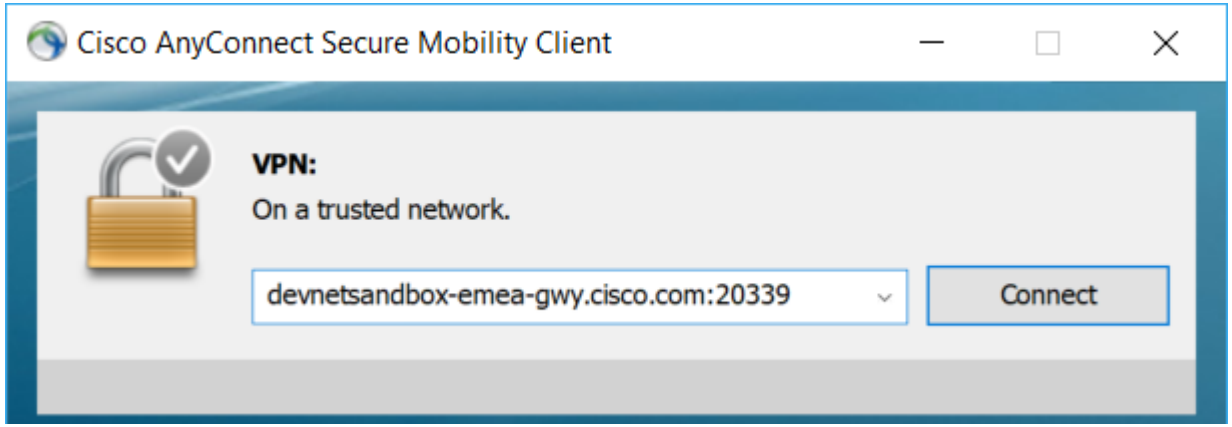
https://developer.cisco.com/site/sandbox/

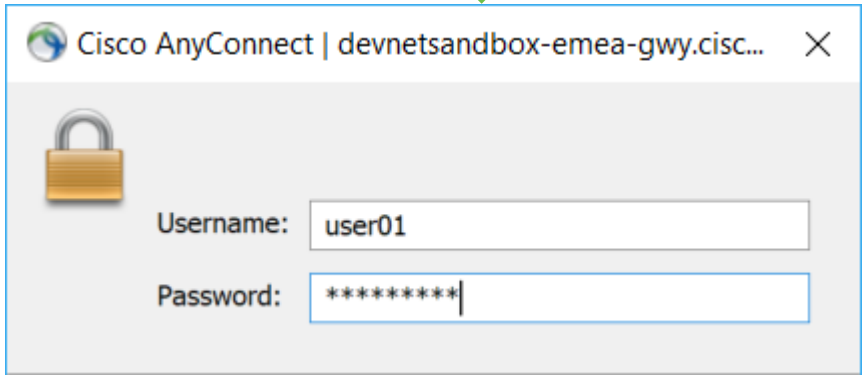
Prototype with the RoomKit Sandboxes





Connect to the DevNet Sandbox





Username **user01,02…**
Password **cisco1234**

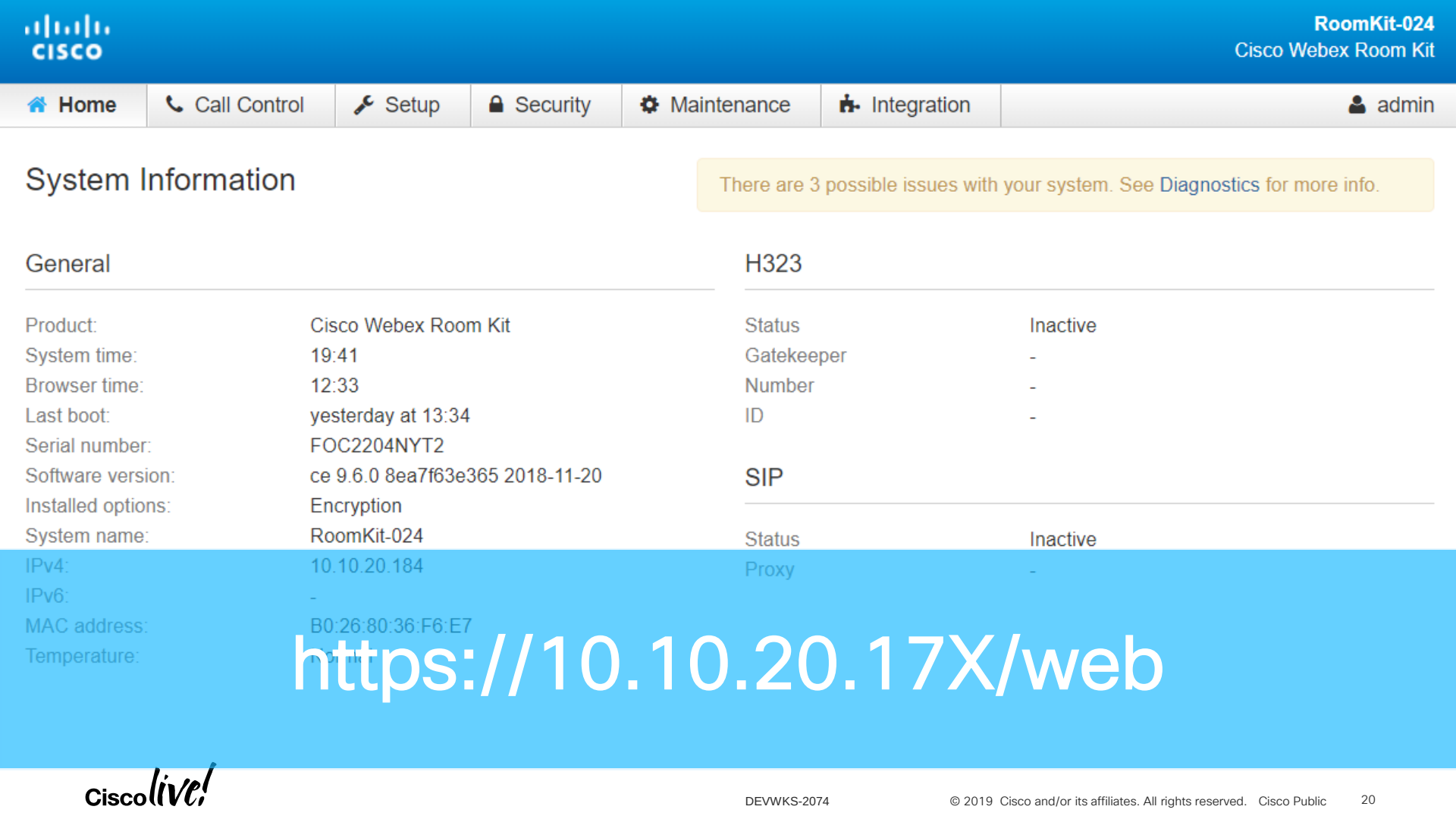
https://10.10.20.171 => https://10.10.20.180

user: **admin**

password: **ciscopsdt**

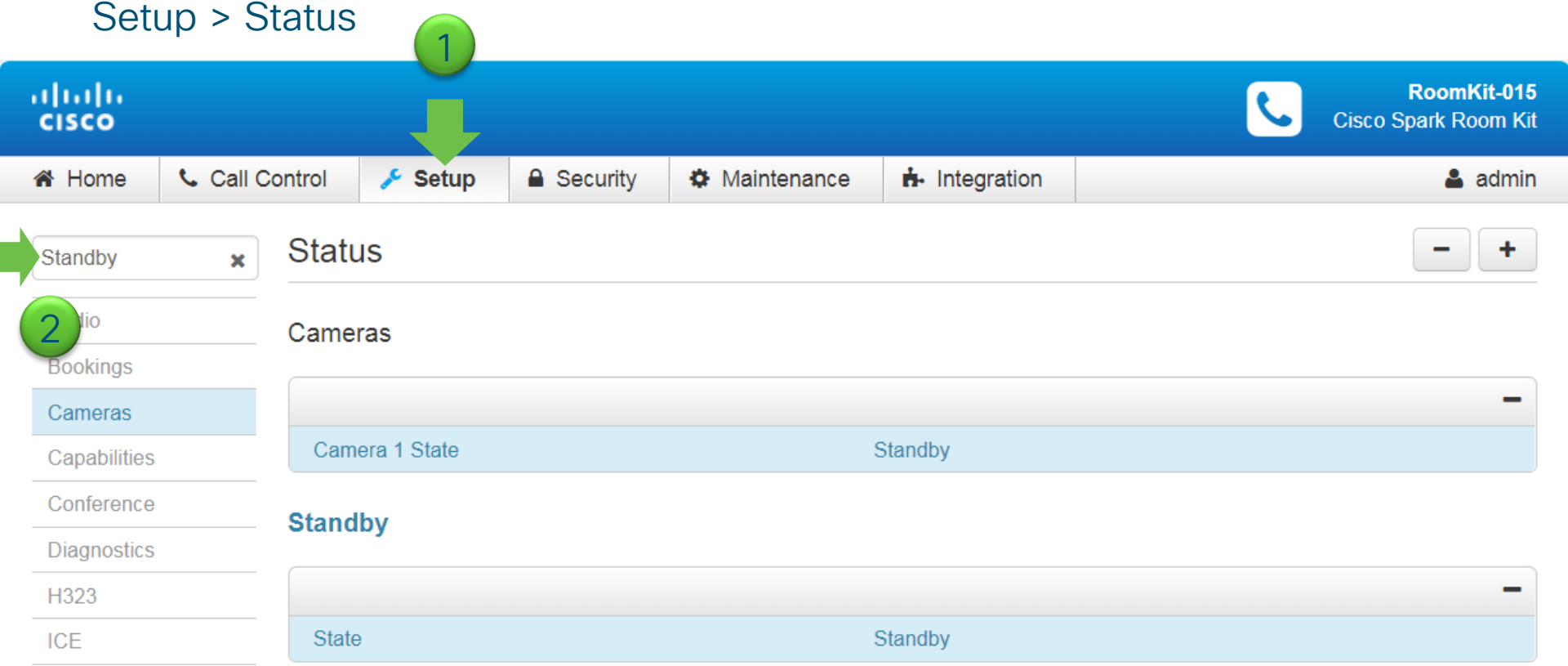
Sandbox VPN: devnetsandbox-emea-gwy.cisco.com:20339

VPN user	passwd	RoomKit IP	user	passwd
user01	cisco1234	https://10.10.20.171	admin	ciscopsdt
user02	cisco1234	https://10.10.20.172	admin	ciscopsdt
user03	cisco1234	https://10.10.20.173	admin	ciscopsdt
user04	cisco1234	https://10.10.20.174	admin	ciscopsdt
user05	cisco1234	https://10.10.20.175	admin	ciscopsdt
user06	cisco1234	https://10.10.20.176	admin	ciscopsdt
user07	cisco1234	https://10.10.20.177	admin	ciscopsdt
user08	cisco1234	https://10.10.20.178	admin	ciscopsdt
user09	cisco1234	https://10.10.20.179	admin	ciscopsdt
user10	cisco1234	https://10.10.20.180	admin	ciscopsdt

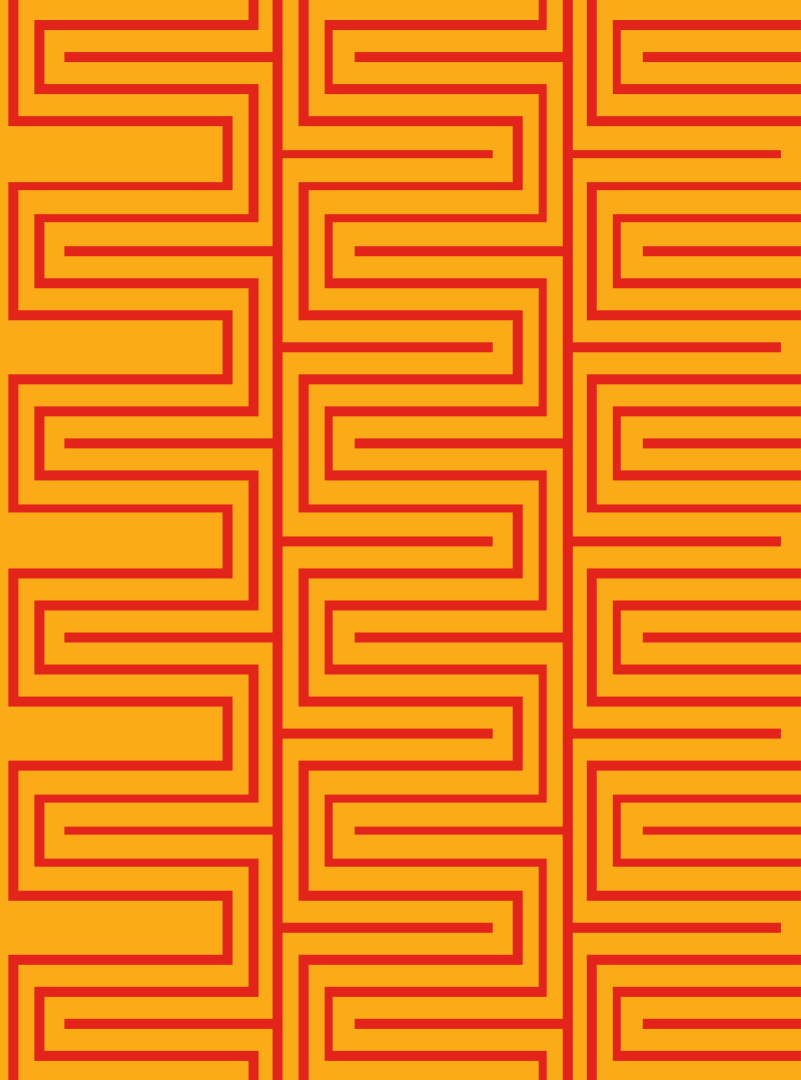


Check your device's Standby State

Setup > Status

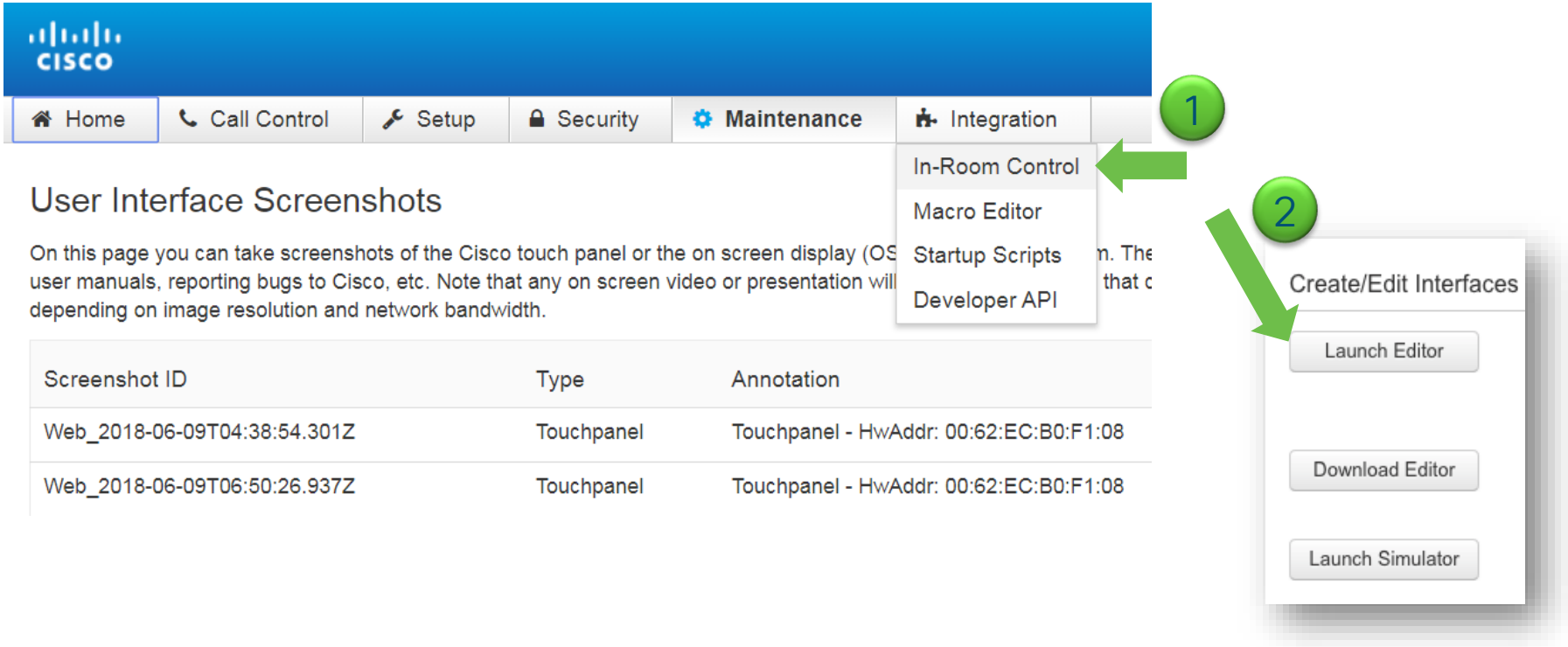


Build a custom In-Room control



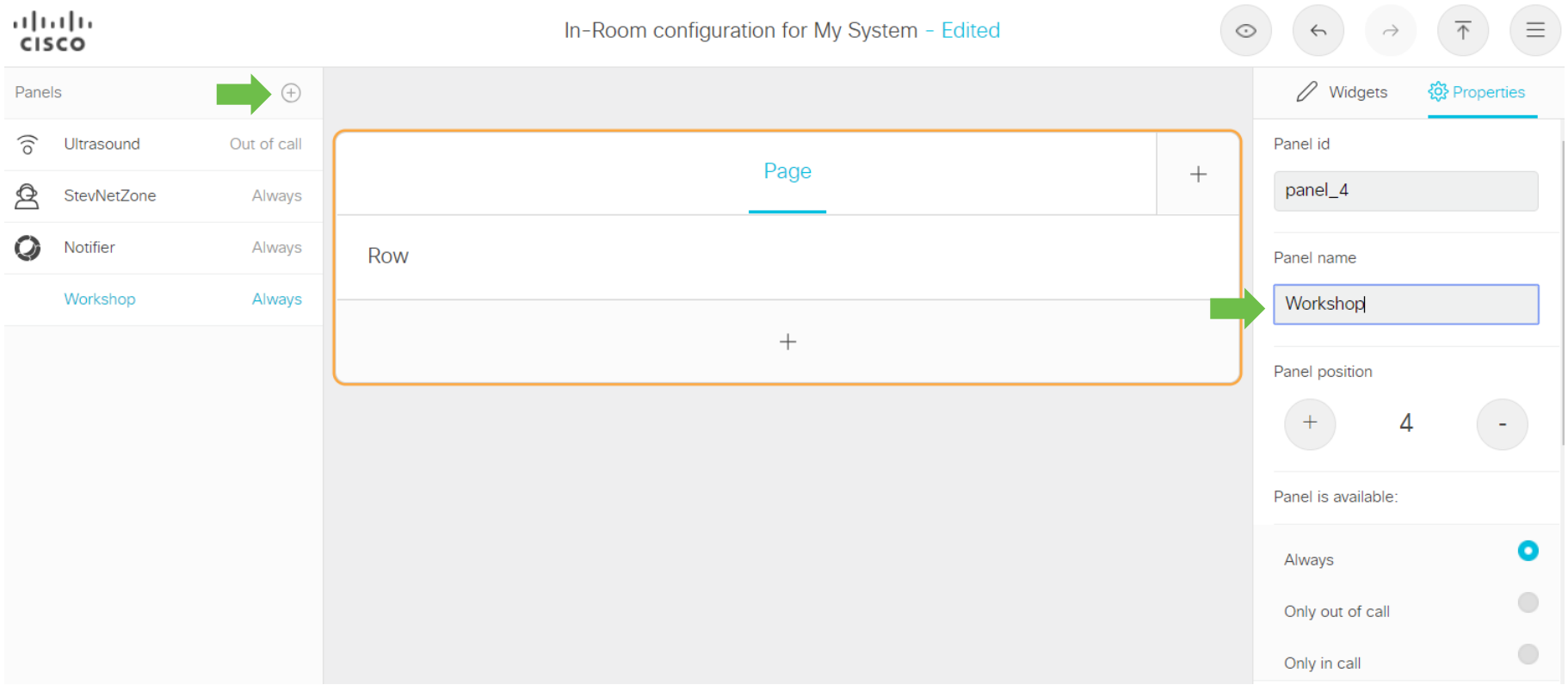
Launch the In-Room Control Editor

https://10.10.20.17X/web



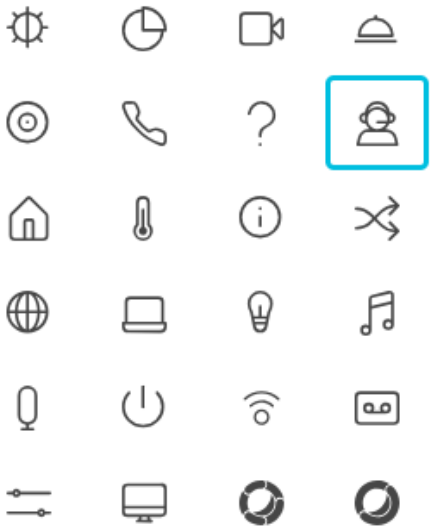
Create a new Panel

https://10.10.20.17x/web/roomcontrol/editor



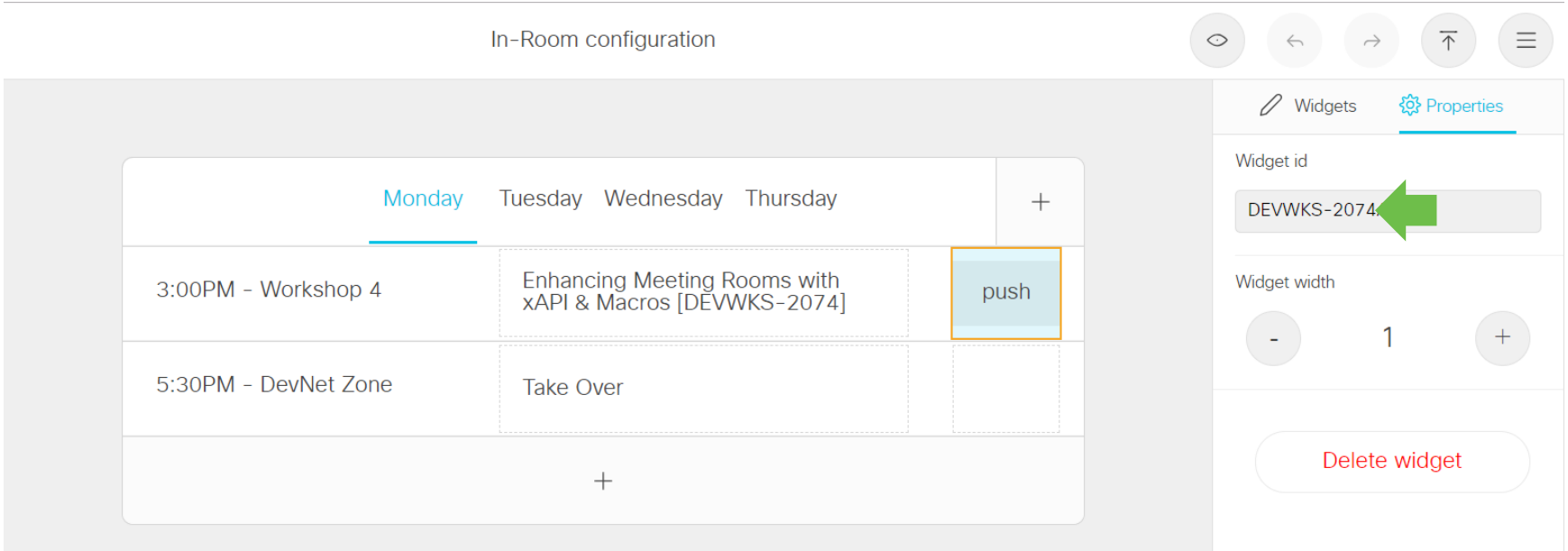
Customize the activity

Icon

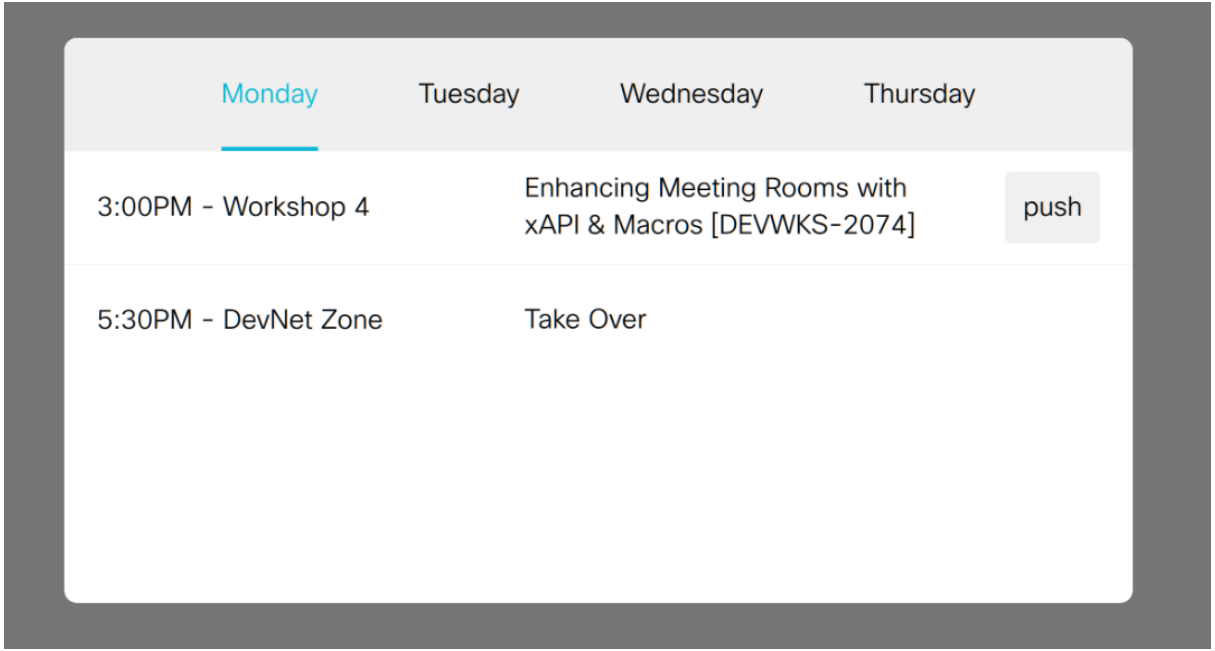


Panel button color

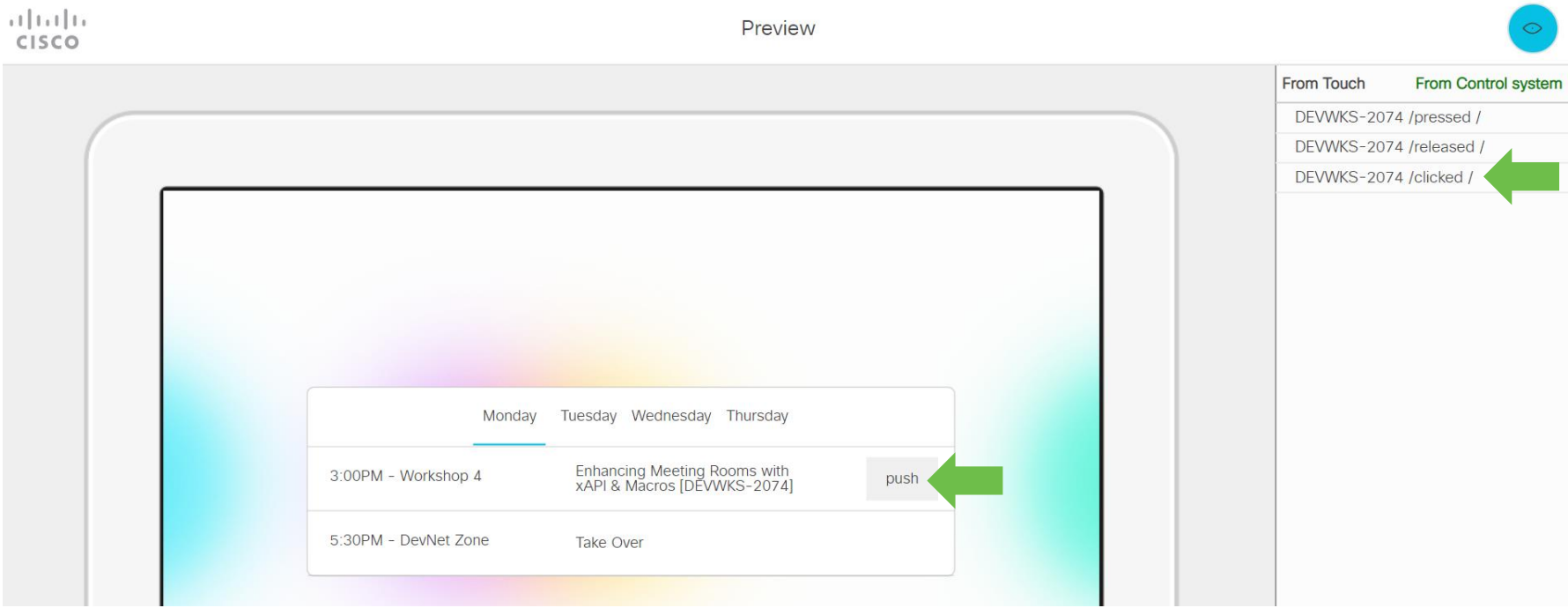
Add a button with id: DEVWKS-2074



Deploy



Preview

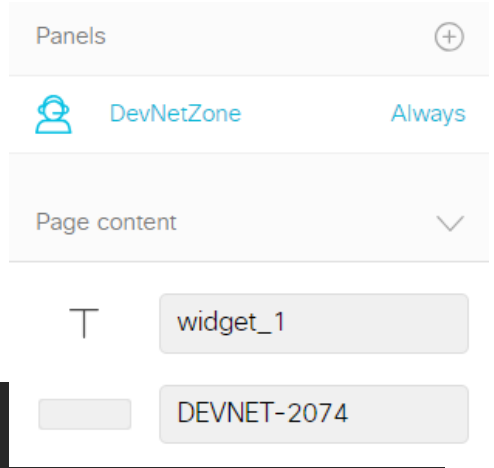


Events fired by custom In-Room Controls

Webex Room Kit capabilities

Pull data (ssh): show widgets

- ssh admin@10.10.20.17x (admin/ciscopsdt)




```
xstatus UserInterface Extensions
*s UserInterface Extensions Widget 1 Value: ""
*s UserInterface Extensions Widget 1 WidgetId: "widget_1"
*s UserInterface Extensions Widget 2 Value: ""
*s UserInterface Extensions Widget 2 WidgetId: "DEVWKS-2074"
** end
```

Listen (ssh): show clicked events

ssh admin@10.10.20.17x

```
xfeedback register /Event/UserInterface/Extensions/Event/Clicked
** end


OK
*e UserInterface Extensions Event Clicked Signal: "DEVWKS-2074"
** end
```

Listen to events in Javascript

- From a Javascript Macro

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

xapi.event.on('UserInterface Extensions Event Clicked Signal', (widgetId) => {
    console.log(`new event from widget: ${widgetId}`)

    // Add your custom logic
})
```

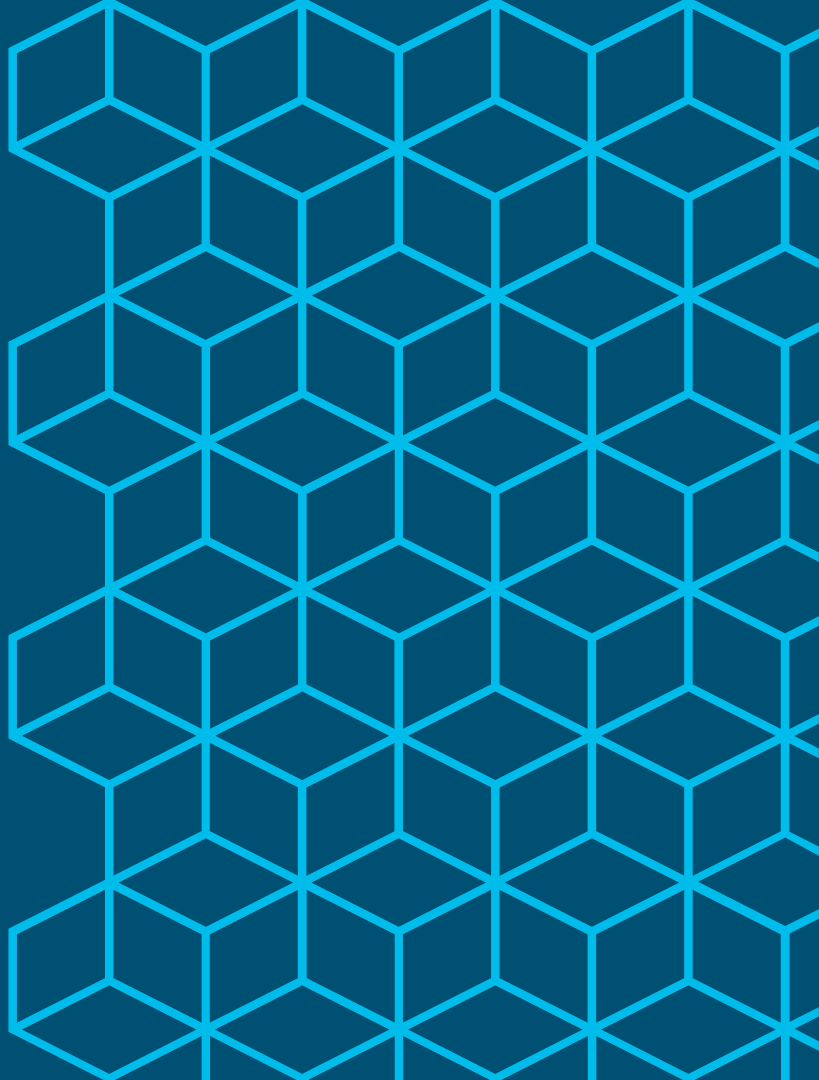
- From the ‘jsxapi’ Node.js module
  https://github.com/ObjectIsAdvantag/xapi-samples/tree/master/jsxapi

/!\ don’t forget to deregister

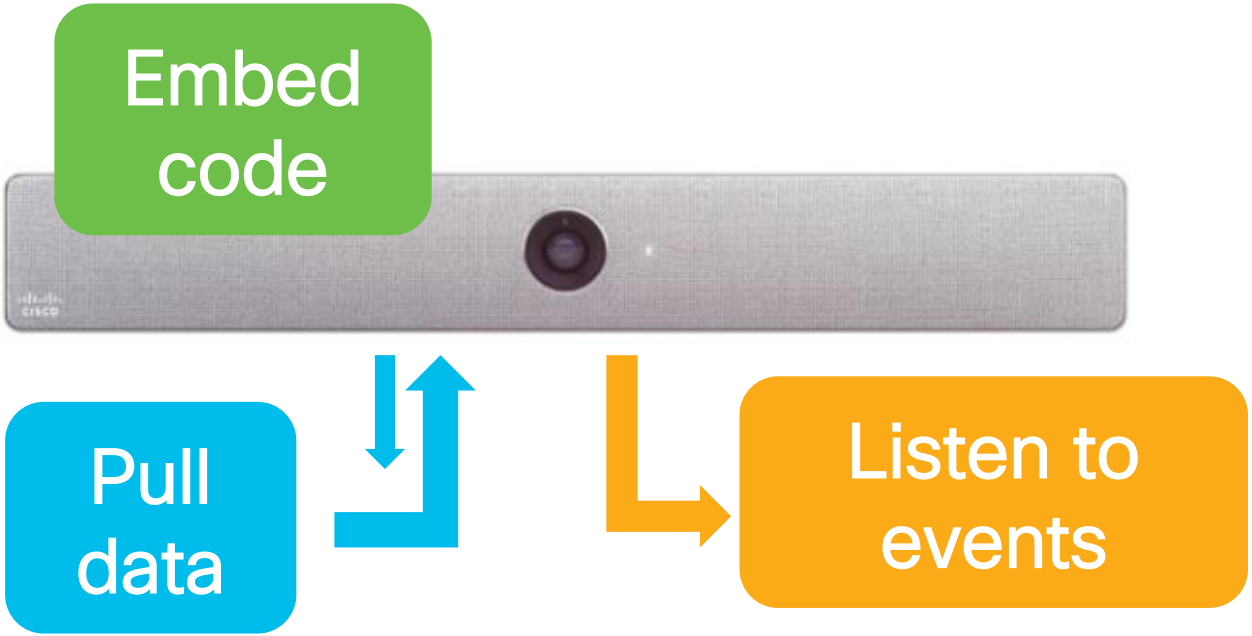
- ssh 10.10.20.17x

```
xfeedback deregisterall
```

Listening to events from a JavaScript Macro



Webex Room Kit capabilities

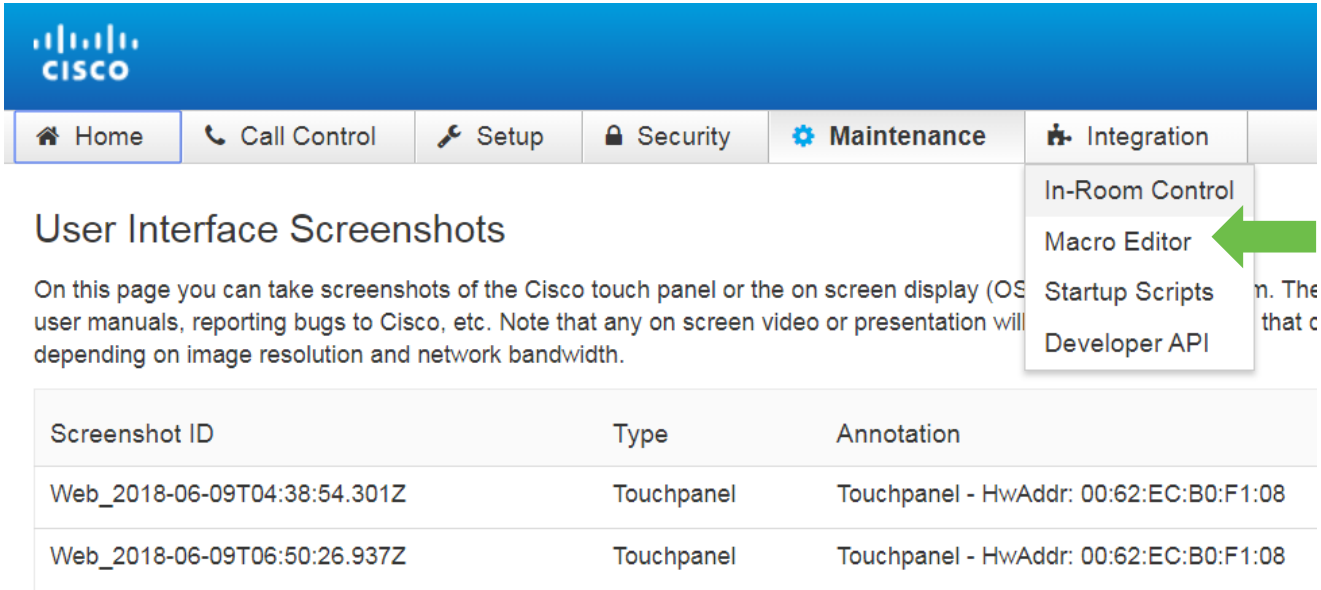


Listening to event clicks via a Macro

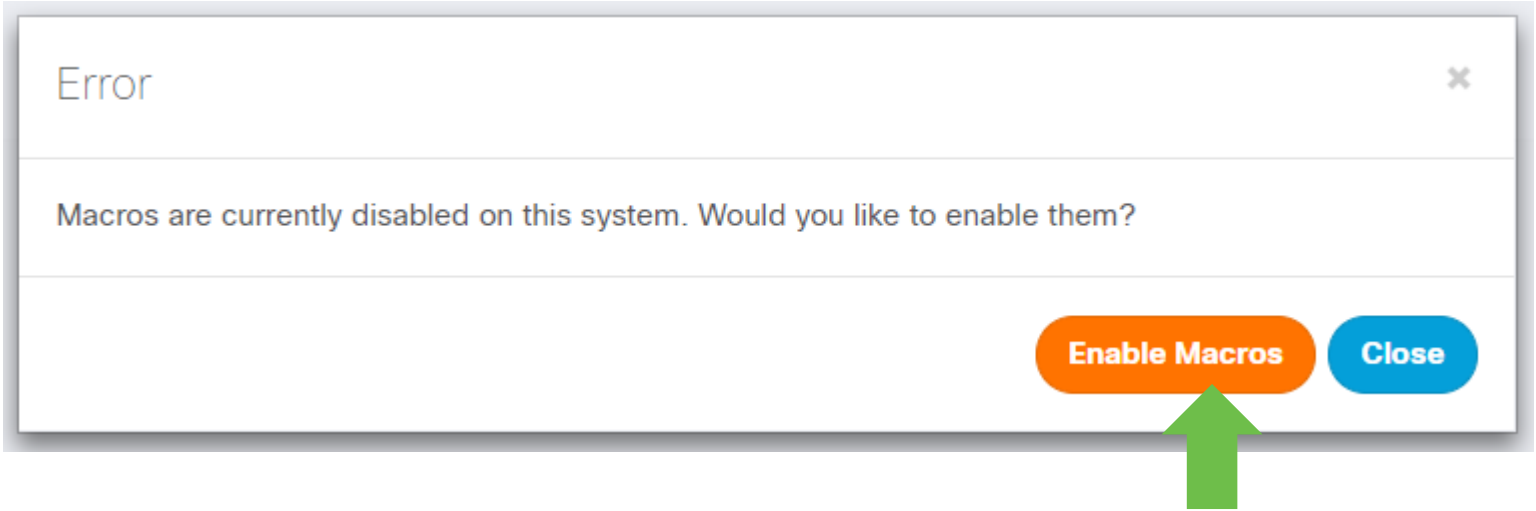
```
const xapi = require('xapi')

xapi.event.on('UserInterface Extensions Event Clicked Signal', (widgetId) => {
    console.log(`new event from widget: ${widgetId}`)

    // Add your custom logic
})

console.log('listening...')
```

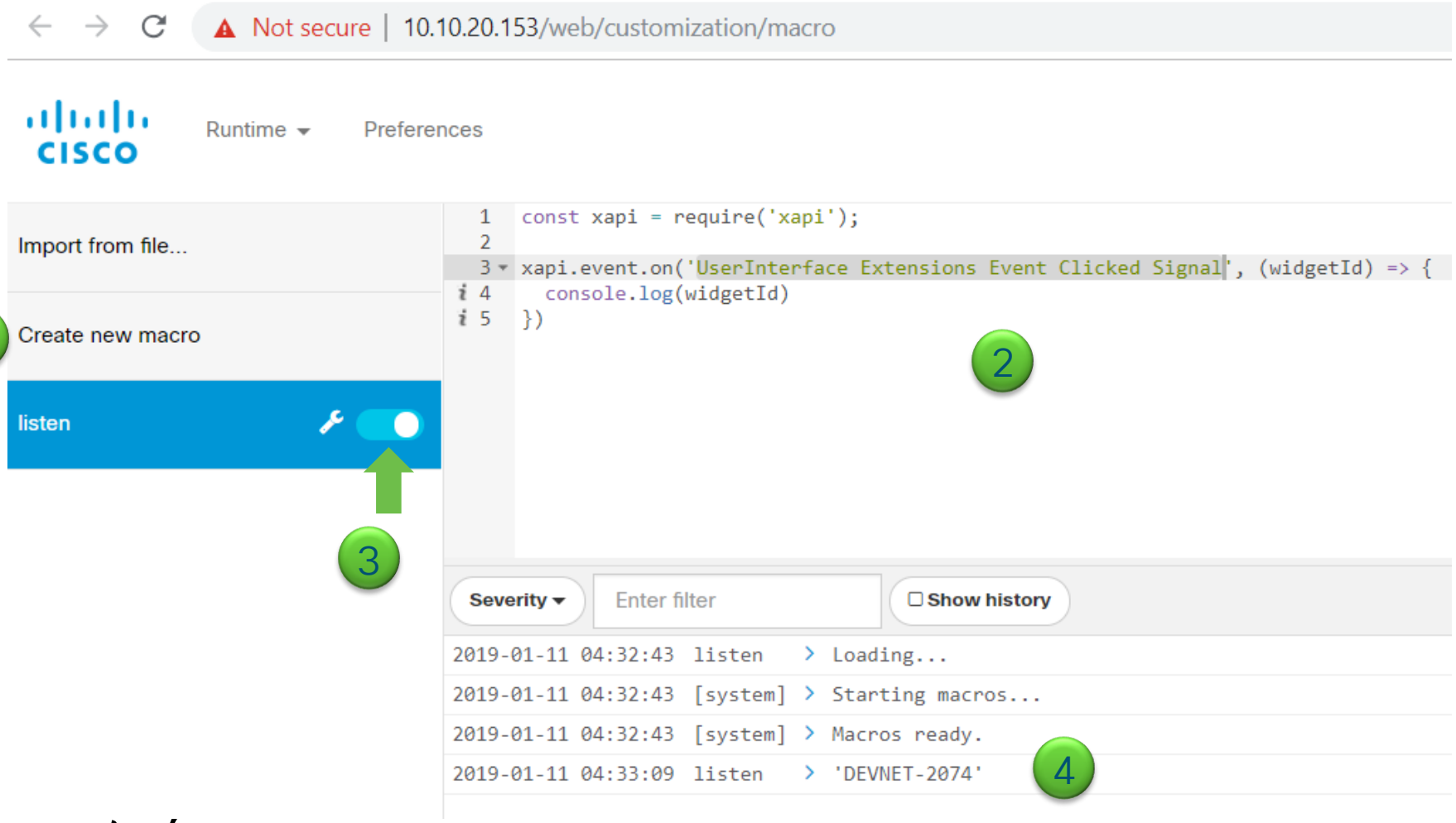
Launch the Macro Editor

https://10.10.20.17X/web

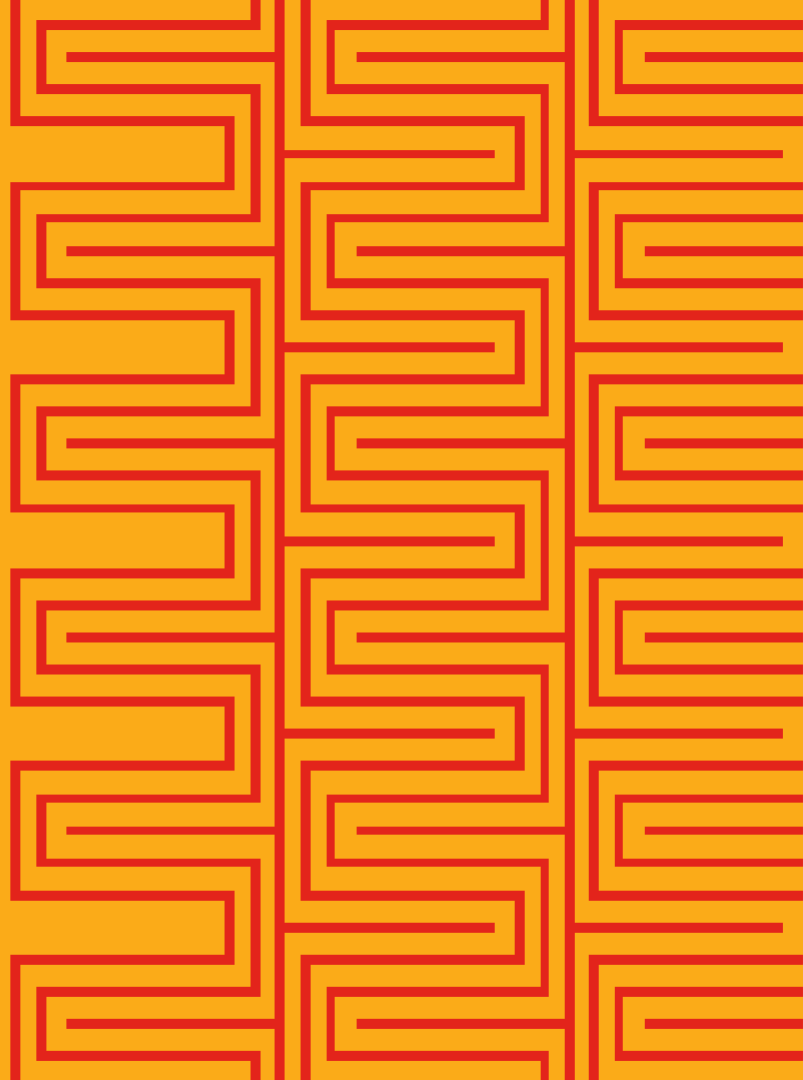


Listening with a Macro



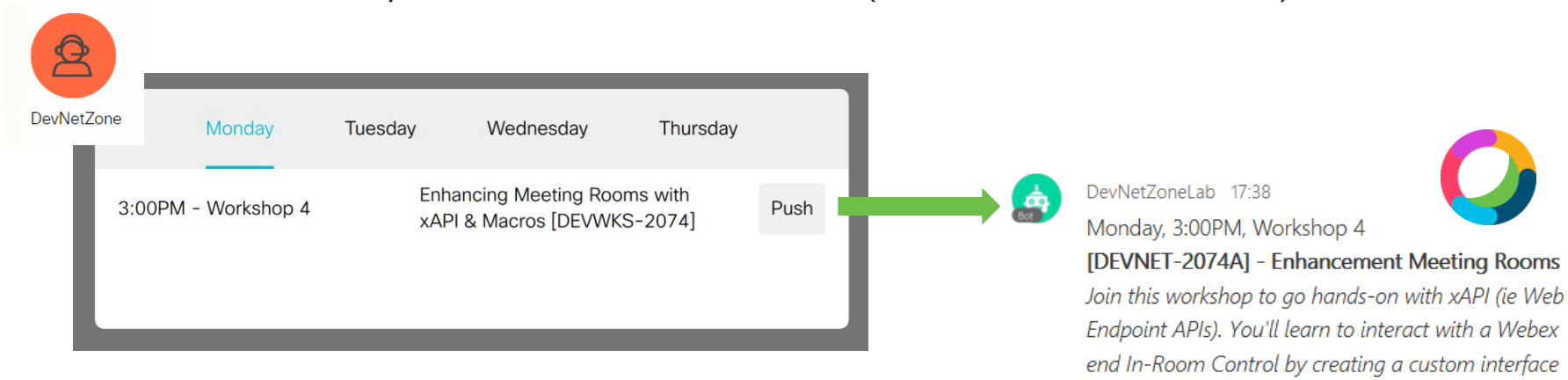


HttpClient example: pushing messages to a Webex Teams space



Our goal for today

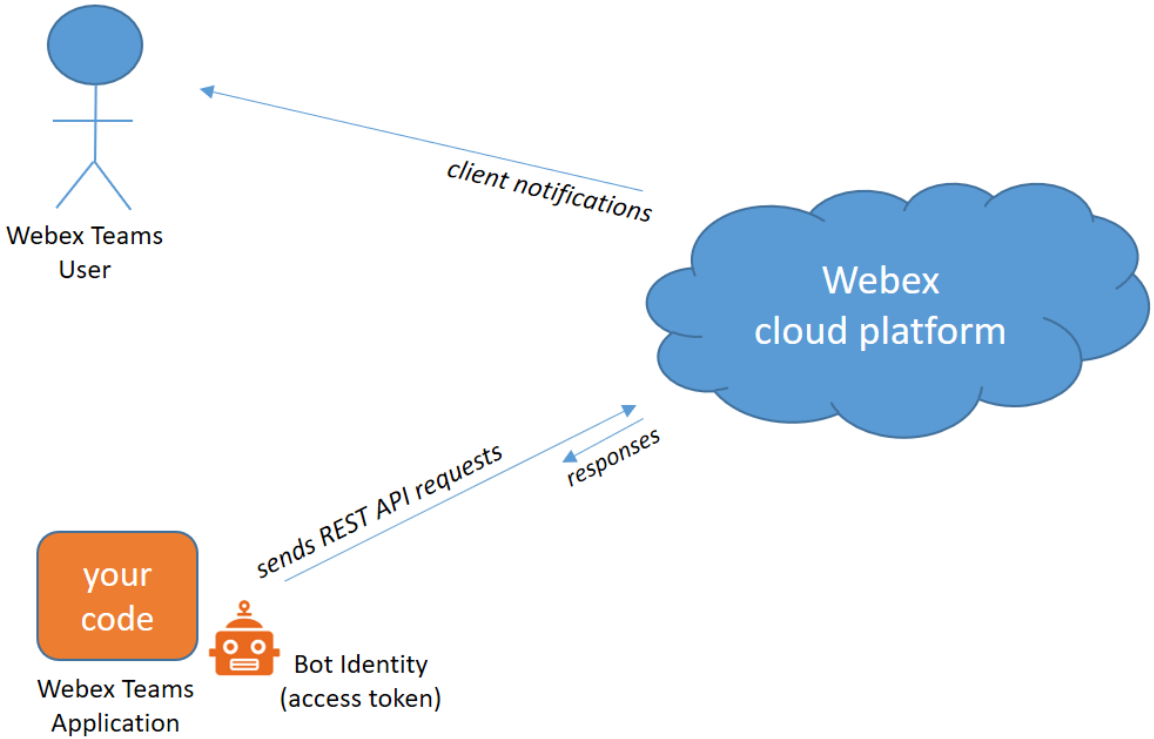
- Connect to a CE 9.6 RoomKit from the DevNet Sandbox
- Build and deploy a custom In-Room Control (CE 9.2.1+)
- Create and run Javascript Macros (CE 9.2.1+)
- Learn the new HttpClient POST command (comes with CE 9.6.1)



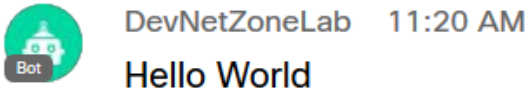
Webex Teams REST API v1

/People	/Rooms	/Membership	/Messages	/Teams	/Webhooks
GET List People	GET List Rooms	GET List Memberships	GET List Messages	GET List Teams	GET List Webhooks
	POST Create a Room	POST Create a Membership	POST Create a Message	POST Create a Team	POST Create a Webhook
GET Get Person Details	GET Get Room details	GET Get Membership details	GET Get Message details	GET Get Team details	GET Get Webhook details
	PUT Update a Room	PUT Update a Membership		PUT Update a Team	PUT Update a Webhook
	DELETE Delete a Room	DELETE Delete a Membership	DELETE Delete a Message	DELETE Delete a Team	DELETE Delete a Webhook

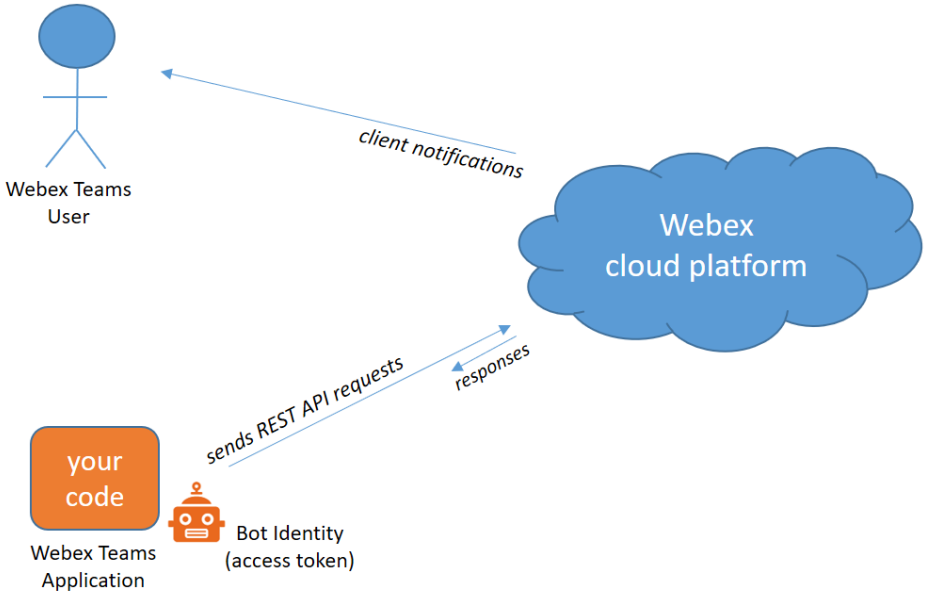
ChatOps: real-time notifications



Big picture




```
48
i 49  push('Hello World', console.log)
50
```



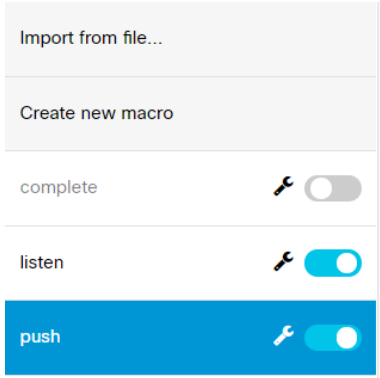
```
function push(msg, cb) {

  // Replace with your bot token
  const token = "MzlkMDU2NzktZWY0OC00MjExLTlhNjItZTFjZDMzMjEzZWU5OTRlODZmZDktZGY2_PF84_adfd15eb-
  // replace with a space your bot is part of
  const roomId = "Y2lzY29zcGFyazovL3VzL1JPT00vYzg0NjlkZDAtMWVmNi0xMWU5LWI1MWYtOTUyNzFiZGM2ZTIz"

  // Post message
  let payload = {
    "markdown": msg,
    "roomId": roomId
  }
  xapi.command(
    'HttpClient Post',
    {
      Header: ["Content-Type: application/json", "Authorization: Bearer " + token],
      Url: "https://api.ciscospark.com/v1/messages",
      AllowInsecureHTTPS: "True"
    },
    JSON.stringify(payload))
    .then((response) => {
      if (response.StatusCode == 200) {
        console.log("message pushed to Webex Teams")
        if (cb) cb(null, response.StatusCode)
        return
      }
```

HttpClient command

- Configuring the HttpClient command via ssh:

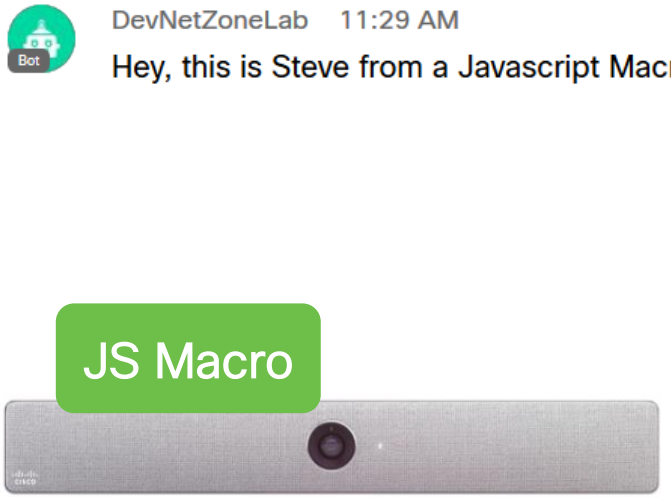
```
xConfiguration HttpClient Mode: On
** end

OK
xConfiguration HttpClient AllowInsecureHTTPS: True
** end

OK
```

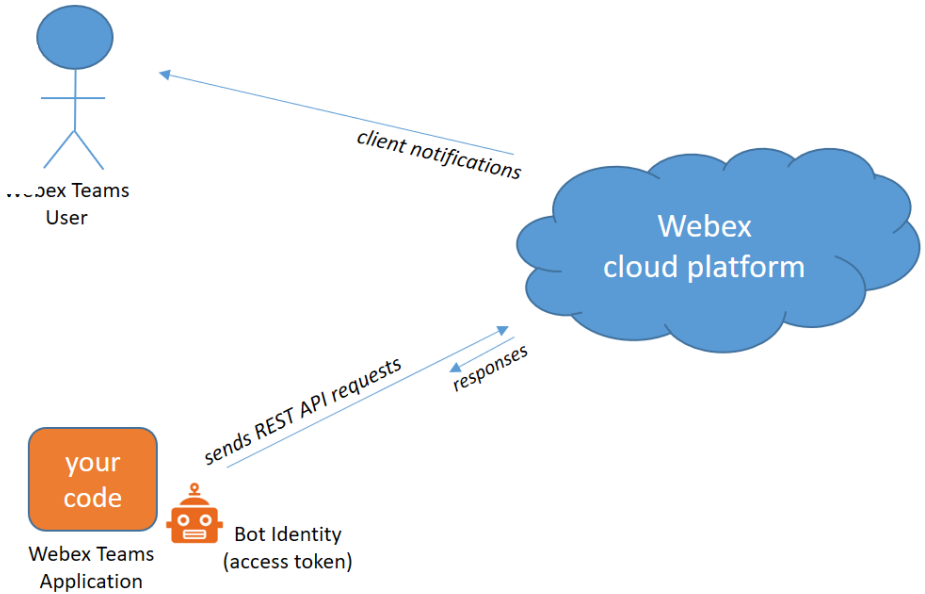
Create the ‘push’ Macro




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

function push(msg, cb) {

  // Replace with your bot token
  const token = "MzlkMDU2NzktZWY0OC00MjExLTlhNjItZTFjZDMzMjEzZWU5OTRlODZmZDktZGY2_PF84_adfd15eb-84e9-4906-b553-94182dee9ade"
  // replace with a space your bot is part of
  const roomId = "Y2lzY29zcGFyazovL3VzL1JPT00vYzg0NjlkZDAtMWVmNi0xMWU5LWI1MWYtOTUyNzFiZGM2ZTIz"

  // Post message
  let payload = {
    "markdown": msg,
    "roomId": roomId
  }
  xapi.command(
    'HttpClient Post',
    {
      Header: ["Content-Type: application/json", "Authorization: Bearer " + token],
      Url: "https://api.ciscospark.com/v1/messages",
      AllowInsecureHTTPS: "True"
    },
    JSON.stringify(payload))
    .then((response) => {
      if (response.StatusCode == 200) {
        console.log("message pushed to Webex Teams")
        if (cb) cb(null, response.StatusCode)
        return
      }

      console.log("failed with status code: " + response.StatusCode)
      if (cb) cb("failed with status code: " + response.StatusCode, response.StatusCode)
    })
    .catch((err) => {
      console.log("failed with err: " + err.message)
      if (cb) cb("Could not post message to Webex Teams")
    })
}

push('Hello World', console.log)
```

Testing time!



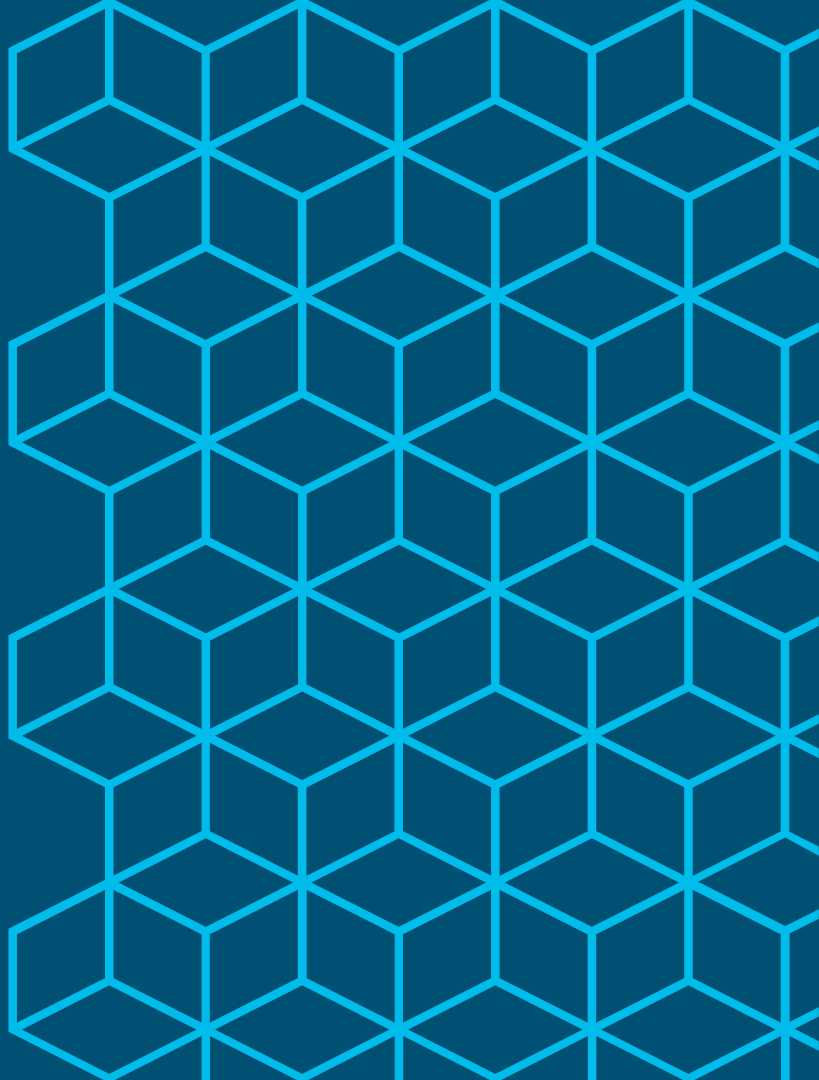



```
48
49  push('Hey, this is Steve from a Javascript Macro', console.log)
50
```

To go further: create your own space

- Create a Bot account for ChatOps purposes
- Copy the bot's access token
- Create a Webex Teams space
- Add the 'roomId@webex.bot' bot to fetch the space identifier
- Add your ChatOps bot to the space
- Update the 'push' Macro with the bot token and space id
- [Optional] Post a message from Postman to check everything looks good

Connecting the dots

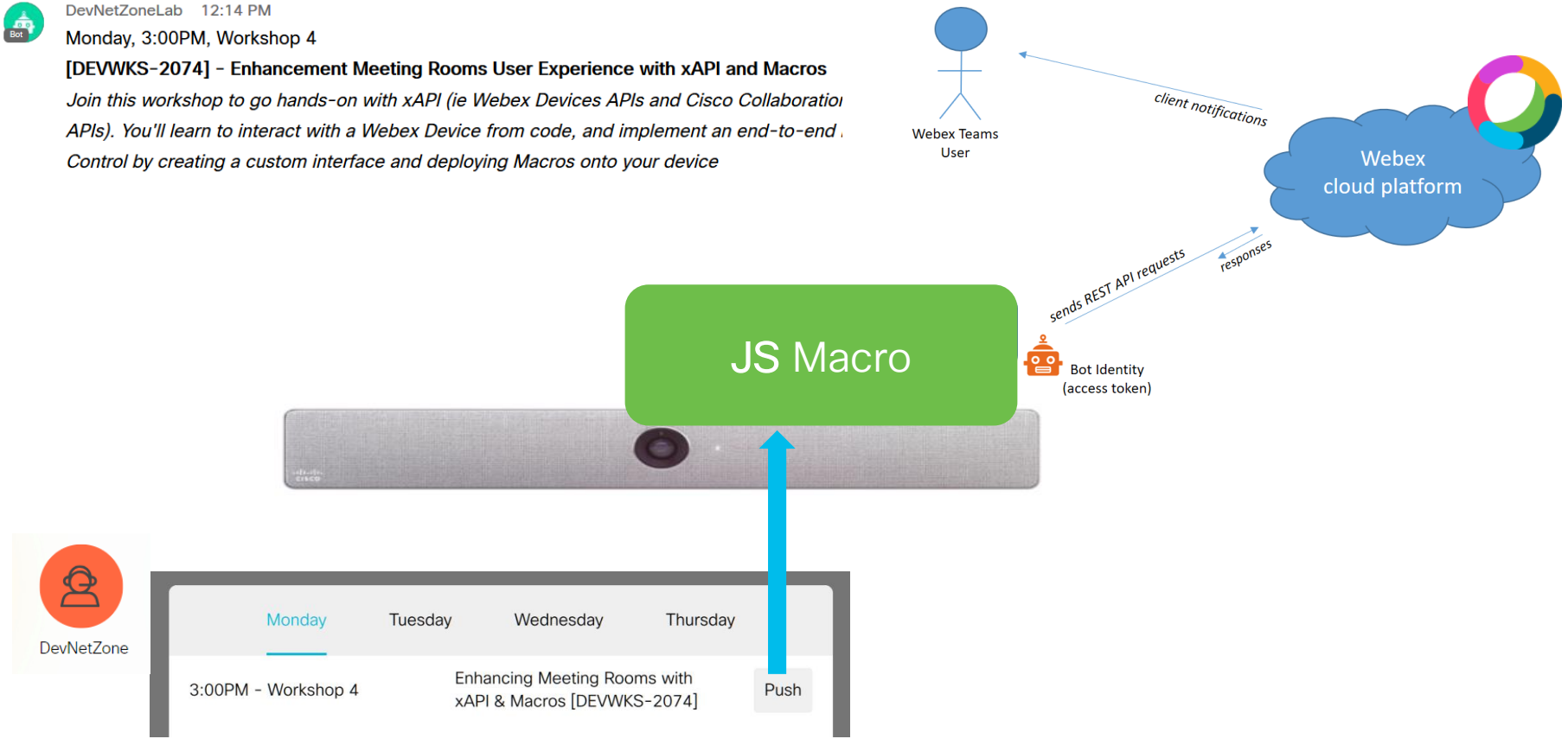


DevNetZoneLab 12:14 PM

Monday, 3:00PM, Workshop 4

[DEVWKS-2074] - Enhancement Meeting Rooms User Experience with xAPI and Macros

Join this workshop to go hands-on with xAPI (ie Webex Devices APIs and Cisco Collaboratio
APIs). You'll learn to interact with a Webex Device from code, and implement an end-to-end
Control by creating a custom interface and deploying Macros onto your device



```
xapi.event.on('UserInterface Extensions Event Clicked Signal', (widgetId) => {
    console.log(`new event from widget: ${widgetId}`)

    let markdown = buildMarkdownForSession(widgetId)
    push(markdown)
})


function buildMarkdownForSession(widgetId) {

    let markdown = `no session found for widget identifier: ${widgetId}`
    let session = sessions[widgetId]
    if (session) {
      console.log(`found session with id: ${widgetId}`)
      markdown = `${session.day}, ${session.time}, ${session.location}`
      markdown += `<br/>**\[${session.id}\] - ${session.title}**`
      markdown += `<br/>_${session.description}_`
    }

    return markdown
}
```

Import from file...

Create new macro

complete

listen

push

```
1  const xapi = require('xapi')
2
3  xapi.event.on('UserInterface Extensions Event Clicked Signal', (widgetId) => {
4      console.log(`new event from widget: ${widgetId}`)
5
6      let markdown = buildMarkdownForSession(widgetId)
7      push(markdown)
8  })
9
10
11 function buildMarkdownForSession(widgetId) {
12
13     let markdown = `no session found for widget identifier: ${widgetId}`
14     let session = sessions[widgetId]
15     if (session) {
16       console.log(`found session with id: ${widgetId}`)
17       markdown = `${session.day}, ${session.time}, ${session.location}`
18       markdown += `<br/>**\[${session.id}\] - ${session.title}**`
19       markdown += `<br/>_${session.description}_`
20     }
21
22     return markdown
23 }
24
25 const sessions = {}
26 sessions['DEVWKS-2074'] = {
27     id: 'DEVWKS-2074',
28     title: "Enhancement Meeting Rooms User Experience with xAPI and Macros",
29     description: "Join this workshop to go hands-on with xAPI (ie Webex Devices APIs and Cisco Collaboration Endpoint APIs). Yo
30     location: "Workshop 4",
31     type: "workshop",
32     day: "Monday"
33
```

Severity | Enter filter | Show history

```
12:16:25  complete  >  Loading...
12:16:25  [system]  >  Starting macros...
12:16:25  [system]  >  Macros ready.
12:16:30  complete  >  'new event from widget: DEVWKS-2074'
12:16:30  complete  >  'found session with id: DEVWKS-2074'
12:16:31  complete  >  'message pushed to Webex Teams'
```

Congrats!!!

DevNetZoneLab 12:16 PM

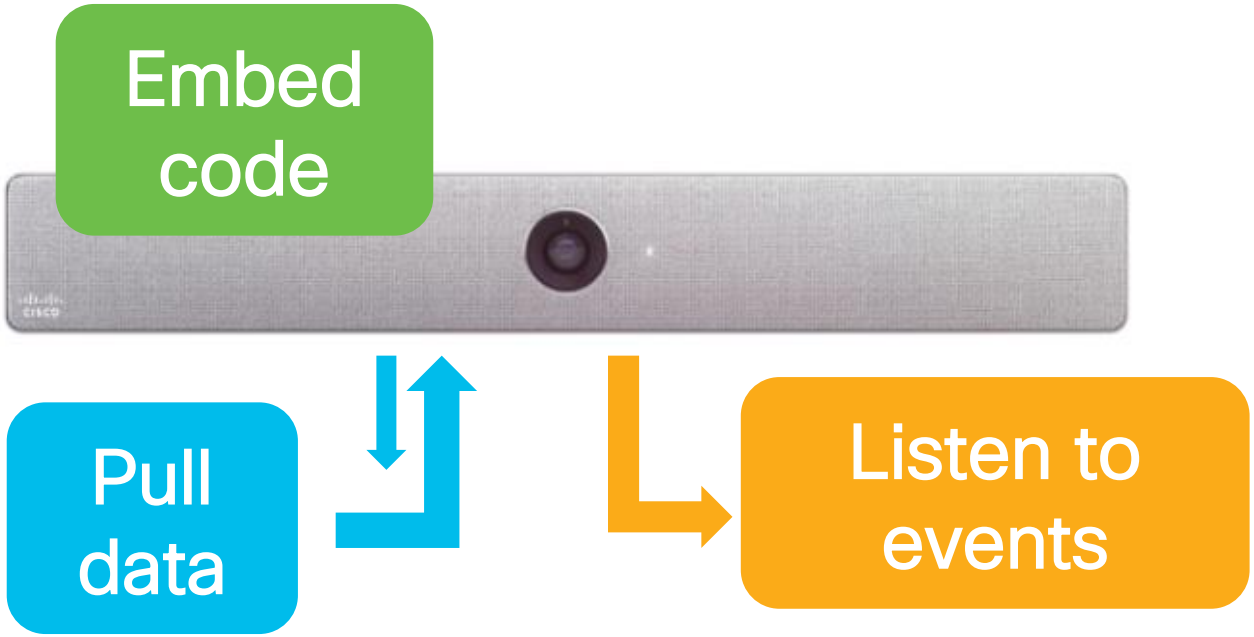
Monday, 3:00PM, Workshop 4

[DEVWKS-2074] - Enhancement Meeting Rooms User Experience with xAPI and Macros

Join this workshop to go hands-on with xAPI (ie Webex Devices APIs and Cisco Collaboration Endpoint APIs). You'll learn to interact with a Webex Device from code, and implement an end-to-end In-Room Control by creating a custom interface and deploying Macros onto your device

Wrapup

Cisco Collaboration Devices Programmability



xAPI, Jan 2019

Javascript Macros

All CE devices ***but SX10***
Admin role *only*
HTTP requests (via HttpClient and HttpFeedback)

Embed code

Pull data

xstatus, xcommand, xconfig

local network connectivity (coming via Webex Cloud)*

Listen to events

xfeedback

Take actions as event fly in from a Control System, 'jsxapi' or a Macro

Inspired? let's continue your journey!

https://github.com/ObjectIsAdvantag/xapi-samples/tree/master/controls

- CLUS Agenda
- User Notifier
- Ultrasound Control
- Maze Game
- In-Room Simulator
- Interactive map

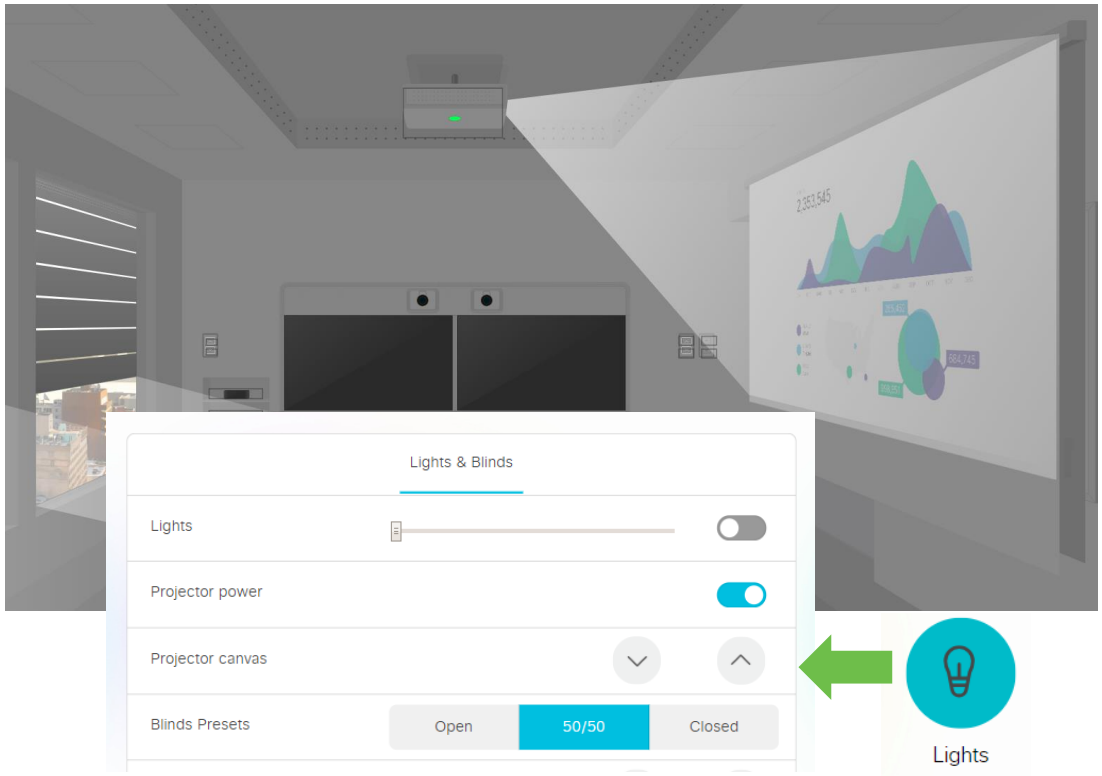
In-Room Simulator

https://controls-editor.herokuapp.com/playground.html?virtualroom

End-to-end enriched User Experience for Meeting Rooms

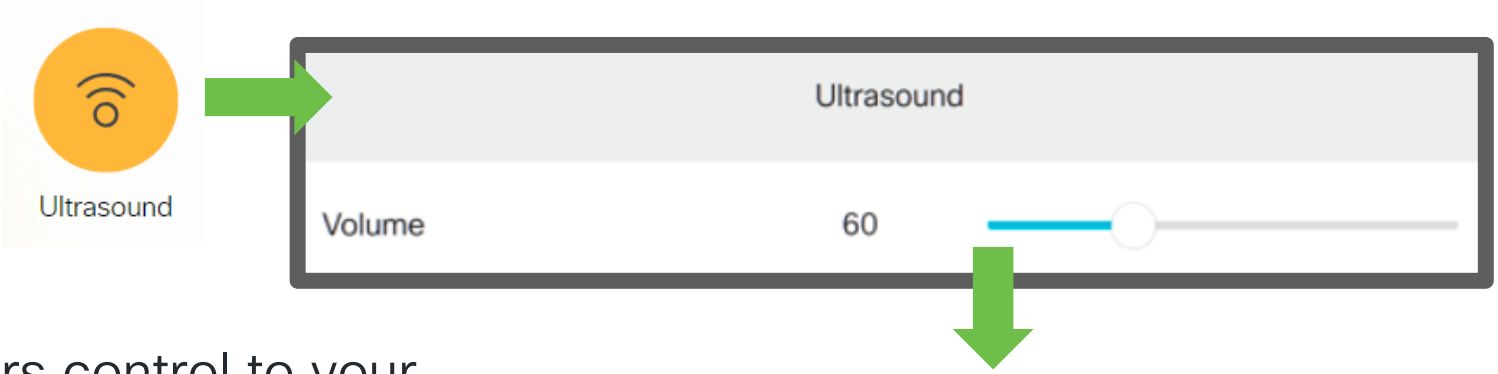
- In-Room Control
- Node.js jsxapi

Note: requires Google Chrome



Ultrasound

https://github.com/ObjectIsAdvantag/xapi-samples/tree/master/controls/ultrasound



Give users control to your devices' Ultrasound MaxVolume

- In-Room Control
- Macro-compatible

/!\ Pairing is disabled

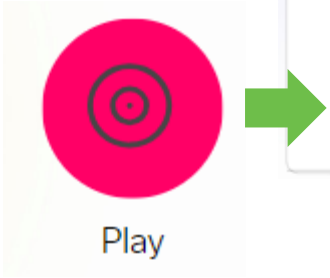
custom message pushed to your devices' Touch10/DX interface as pairing is enabled / disabled.

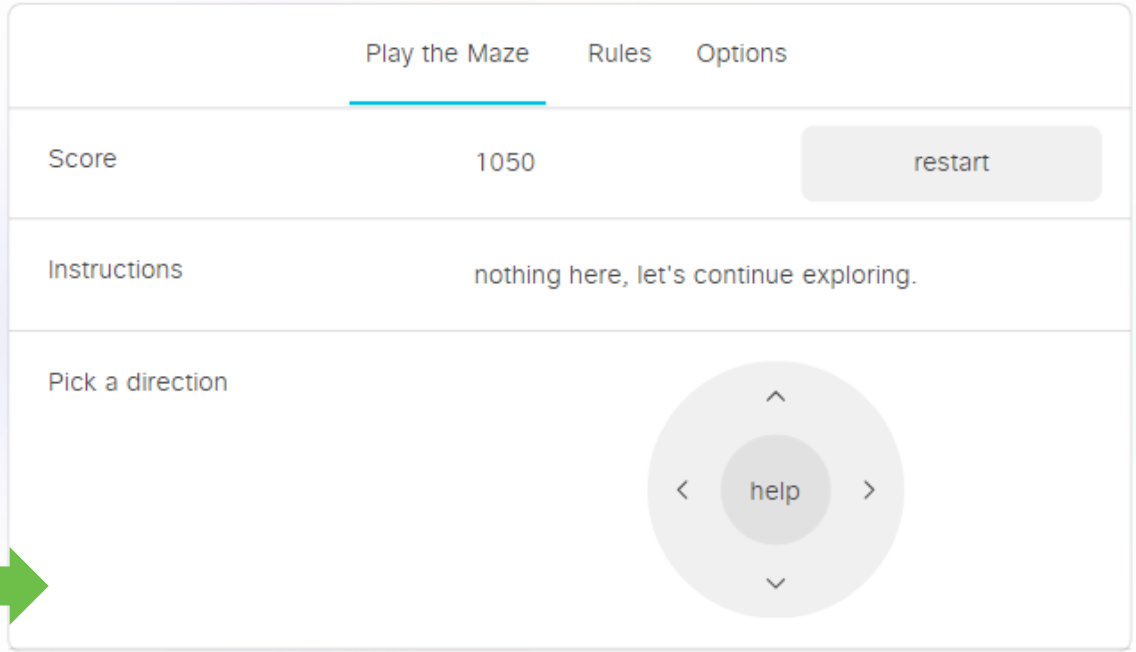
Maze Game

https://github.com/ObjectIsAdvantag/xapi-samples/tree/master/controls/maze-scores

End-to-end enriched User Experience for Meeting Rooms

- In-Room Control
- Macro





xAPI module at DevNet

https://learninglabs.cisco.com/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 20 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 and Macros

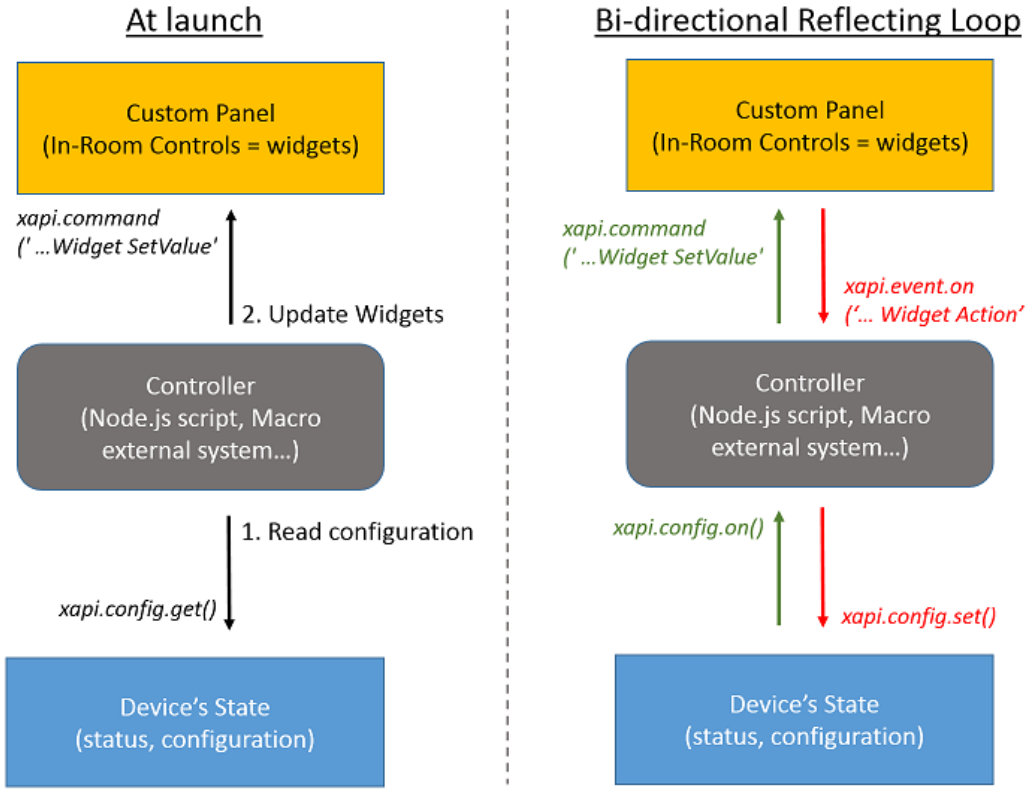
Learn how to create custom controls for Cisco Collaboration Devices, using the on-board control simulator tool and the in-room control editor. Then make those controls interactive via a server-based Node.js script, or an on-board JavaScript macro.

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.

In-Room Control state management

https://learninglabs.cisco.com/lab/collab-xapi-controls/step/6



Developer Resources

- xAPI learning module
  - https://learninglabs.cisco.com/modules/xapi-intro
- xAPI samples
  - https://github.com/ObjectIsAdvantag/xapi-samples
- ‘xAPI devs’ community space
  - http://bit.ly/join-xapi-devs
- awesome-xapi: curated list of developer resources
  - code samples, sandboxes, labs…
  - https://github.com/CiscoDevNet/awesome-xapi

Cisco DevNet Code Exchange

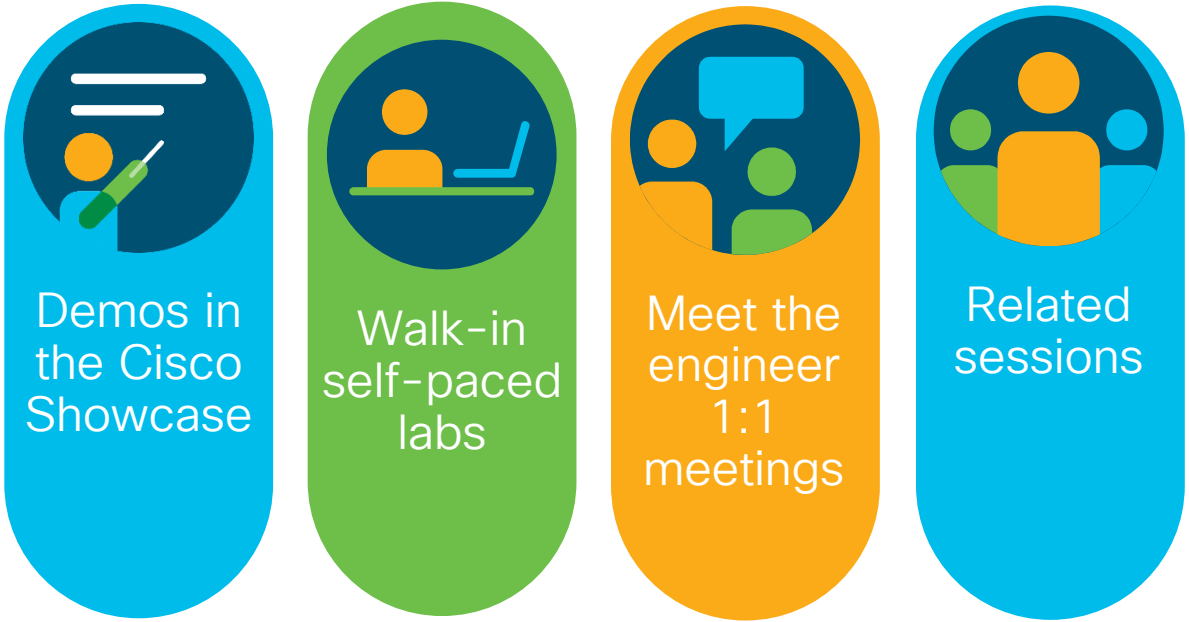
Share your code

Build your developer cred

Get your code in front of the DevNet Community

developer.cisco.com/codeexchange

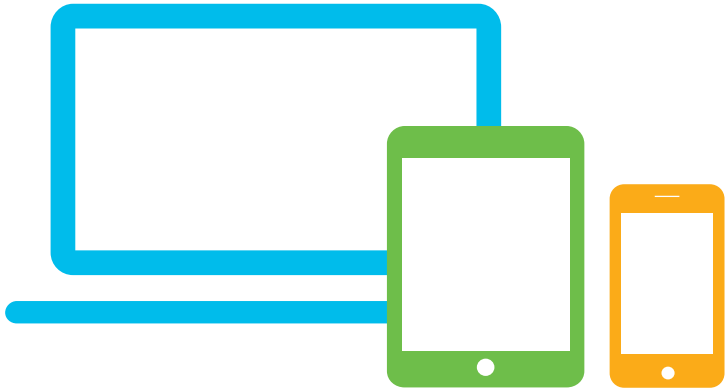
Continue Your Education



Complete your online session survey

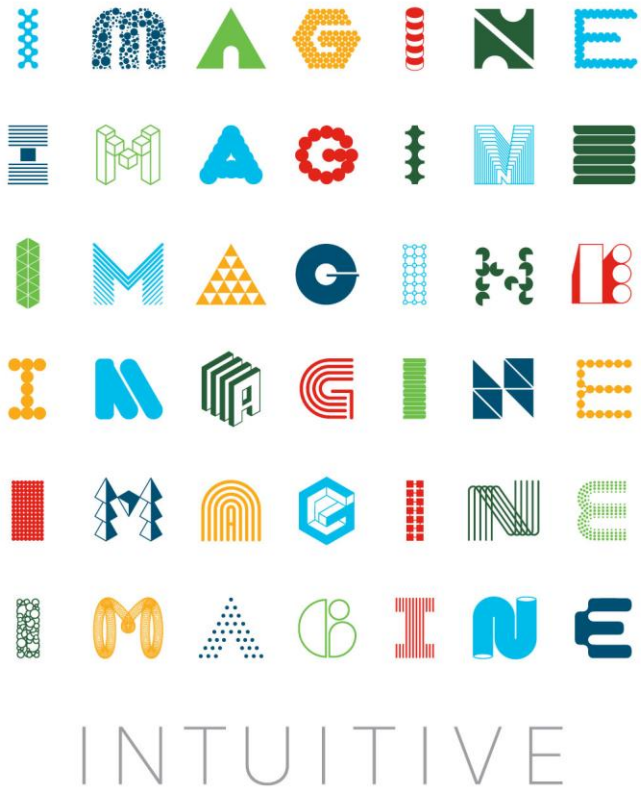
- Please complete your Online Session Survey after each session
- Complete 4 Session Surveys & the Overall Conference Survey (available from Thursday) to receive your Cisco Live T-shirt
- All surveys can be completed via the Cisco Events Mobile App or the Communication Stations

Don’t forget: Cisco Live sessions will be available for viewing on demand after the event at ciscolive.cisco.com



Cisco live!

Thank you



INTUITIVE