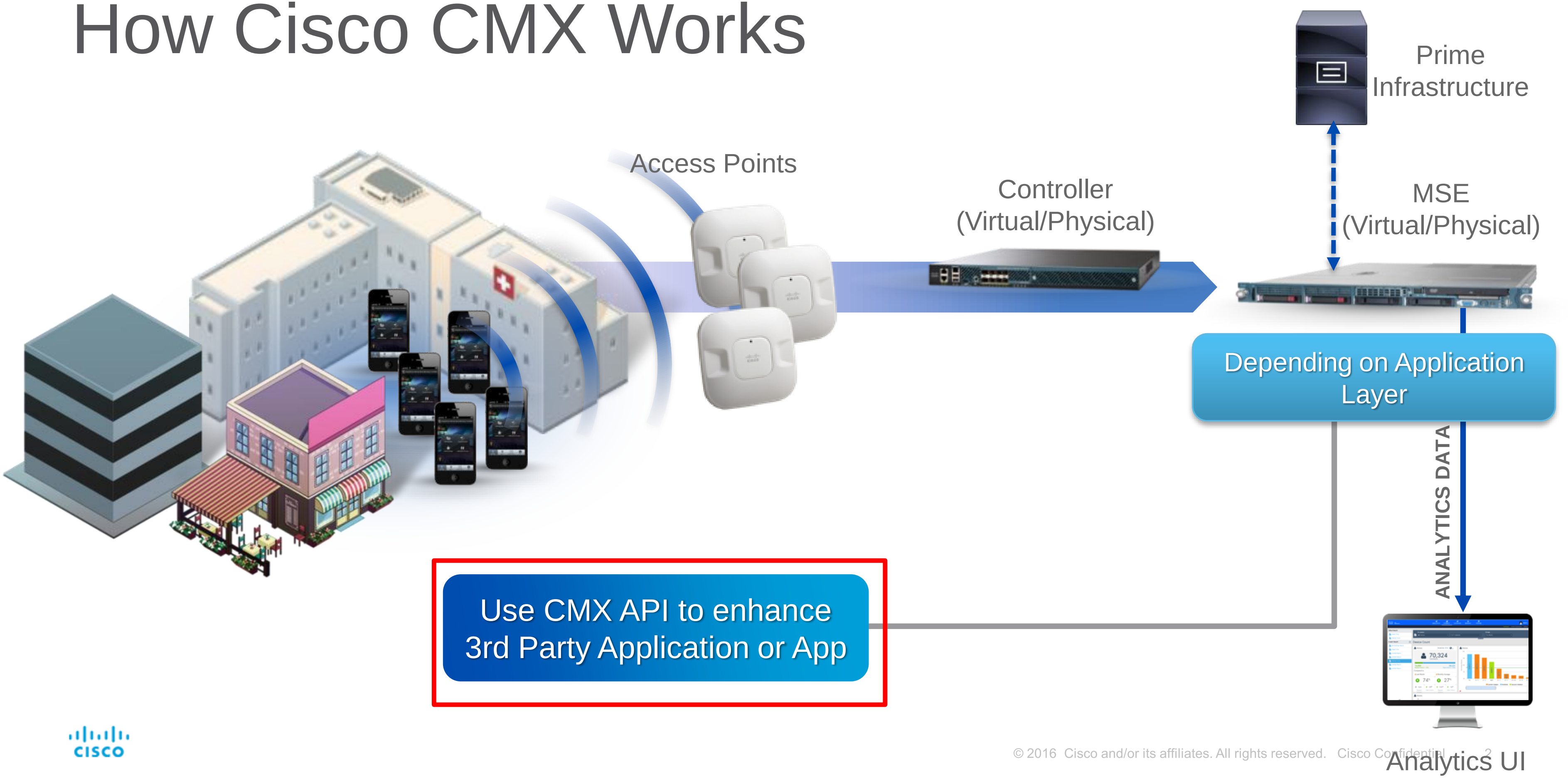
DevNet Coffee Break
CMX

Cisco Connect Croatia 2017

Djordje Vulovic
Consulting Systems Engineer
dvulovic@cisco.com

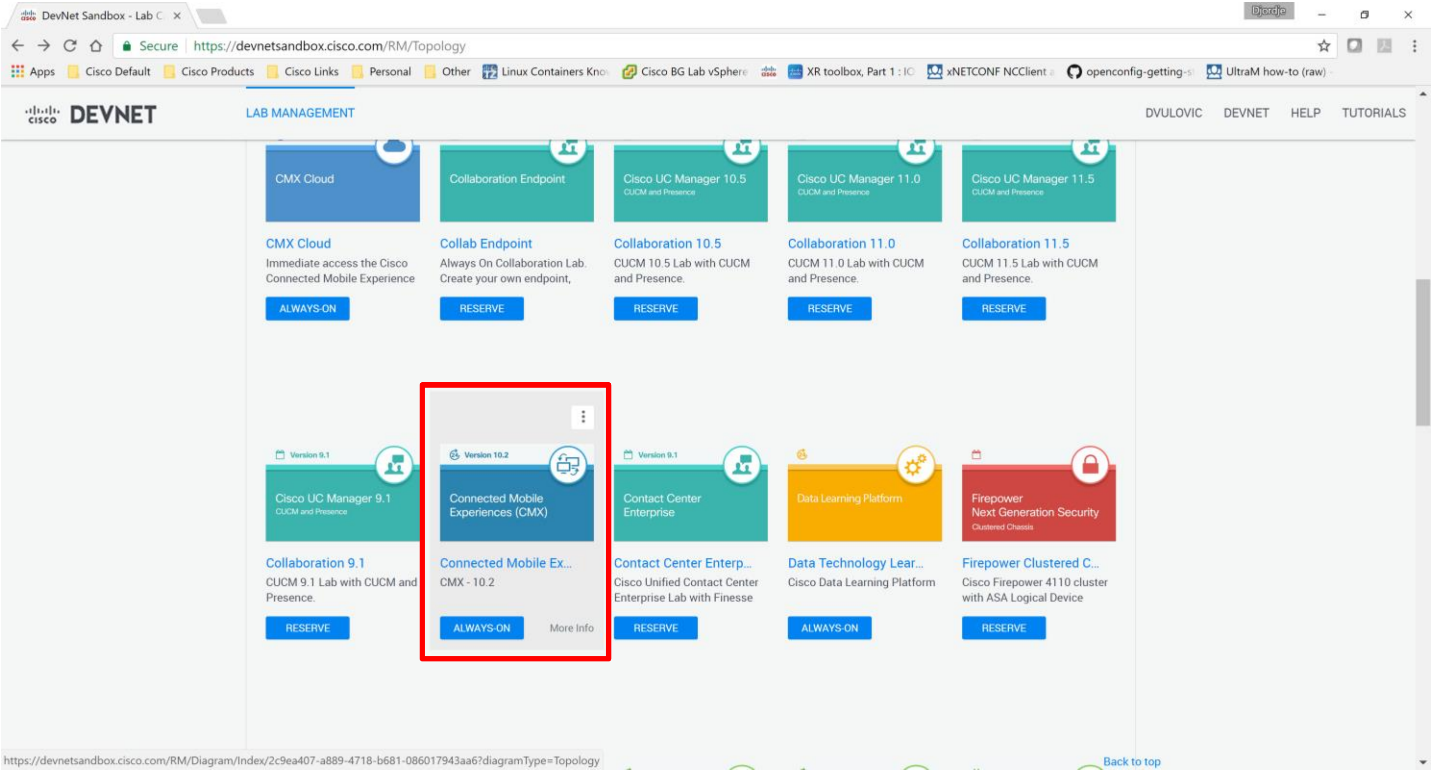
https://github.com/djordjevulovic/Cisco_Connect_HR_2017

How Cisco CMX Works



Use CMX API to enhance 3rd Party Application or App

DevNet Sandbox Labs



CMX Cloud

CMX Cloud

Immediate access the Cisco Connected Mobile Experience

ALWAYS-ON

Collaboration Endpoint

Collab Endpoint

Always On Collaboration Lab. Create your own endpoint,

RESERVE

Cisco UC Manager 10.5
CUCM and Presence

Collaboration 10.5

CUCM 10.5 Lab with CUCM and Presence.

RESERVE

Cisco UC Manager 11.0
CUCM and Presence

Collaboration 11.0

CUCM 11.0 Lab with CUCM and Presence.

RESERVE

Cisco UC Manager 11.5
CUCM and Presence

Collaboration 11.5

CUCM 11.5 Lab with CUCM and Presence.

RESERVE

Version 9.1
Cisco UC Manager 9.1
CUCM and Presence

Collaboration 9.1

CUCM 9.1 Lab with CUCM and Presence.

RESERVE

Version 10.2
Connected Mobile Experiences (CMX)

Connected Mobile Ex...

CMX - 10.2

ALWAYS-ON More Info

Version 9.1
Contact Center Enterprise

Contact Center Enterp...

Cisco Unified Contact Center Enterprise Lab with Finesse

RESERVE

Data Learning Platform

Data Technology Lear...

Cisco Data Learning Platform

ALWAYS-ON

Firepower
Next Generation Security
Clustered Chassis

Firepower Clustered C...

Cisco Firepower 4110 cluster with ASA Logical Device

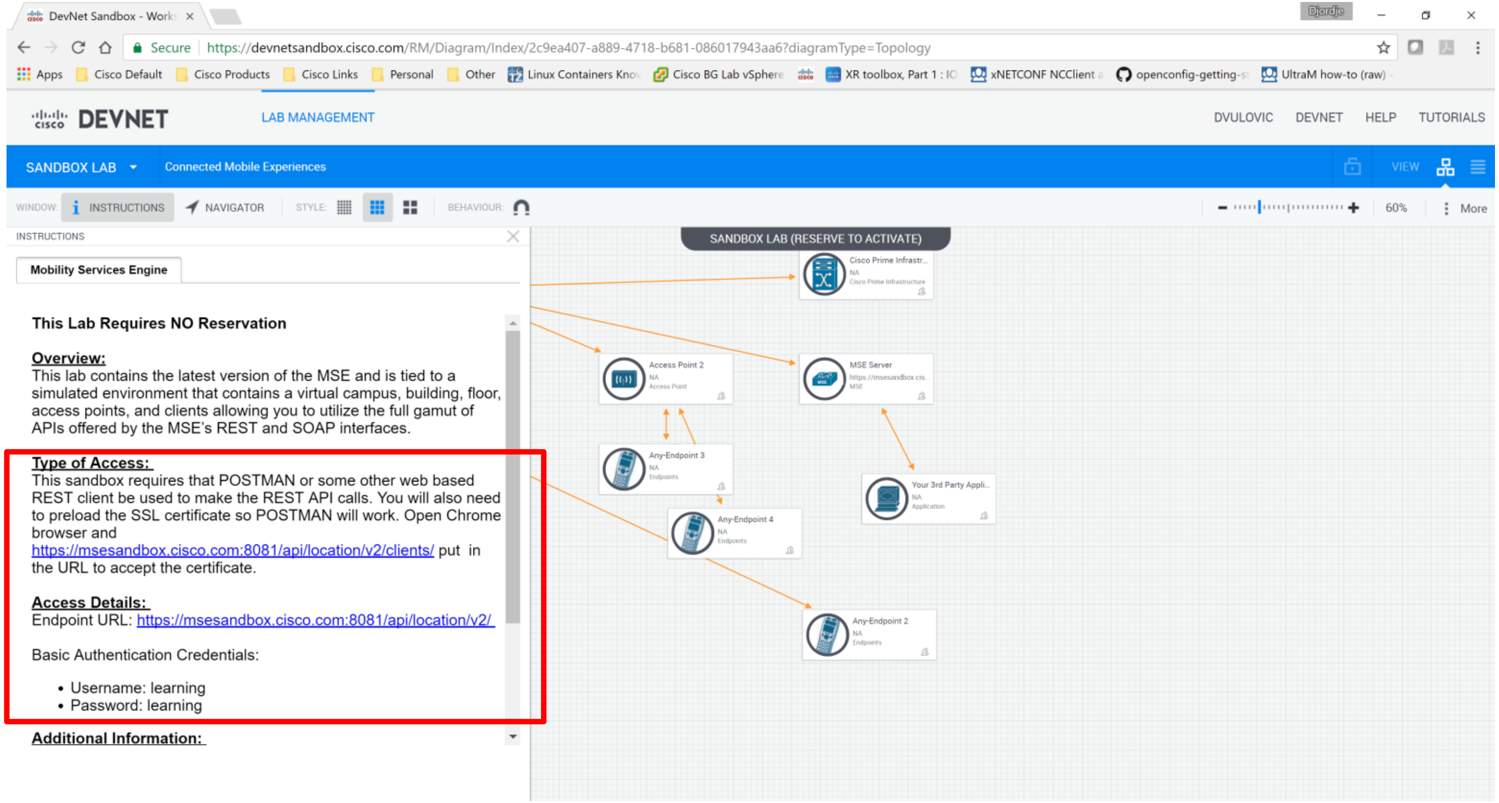
RESERVE

https://devnetsandbox.cisco.com/RM/Diagram/Index/2c9ea407-a889-4718-b681-086017943aa6?diagramType=Topology

Back to top

https://devnetsandbox.cisco.com

DevNet CMX SandBox



INSTRUCTIONS

Mobility Services Engine

This Lab Requires NO Reservation

Overview:
This lab contains the latest version of the MSE and is tied to a simulated environment that contains a virtual campus, building, floor, access points, and clients allowing you to utilize the full gamut of APIs offered by the MSE's REST and SOAP interfaces.

Type of Access:
This sandbox requires that POSTMAN or some other web based REST client be used to make the REST API calls. You will also need to preload the SSL certificate so POSTMAN will work. Open Chrome browser and https://msesandbox.cisco.com:8081/api/location/v2/clients/ put in the URL to accept the certificate.

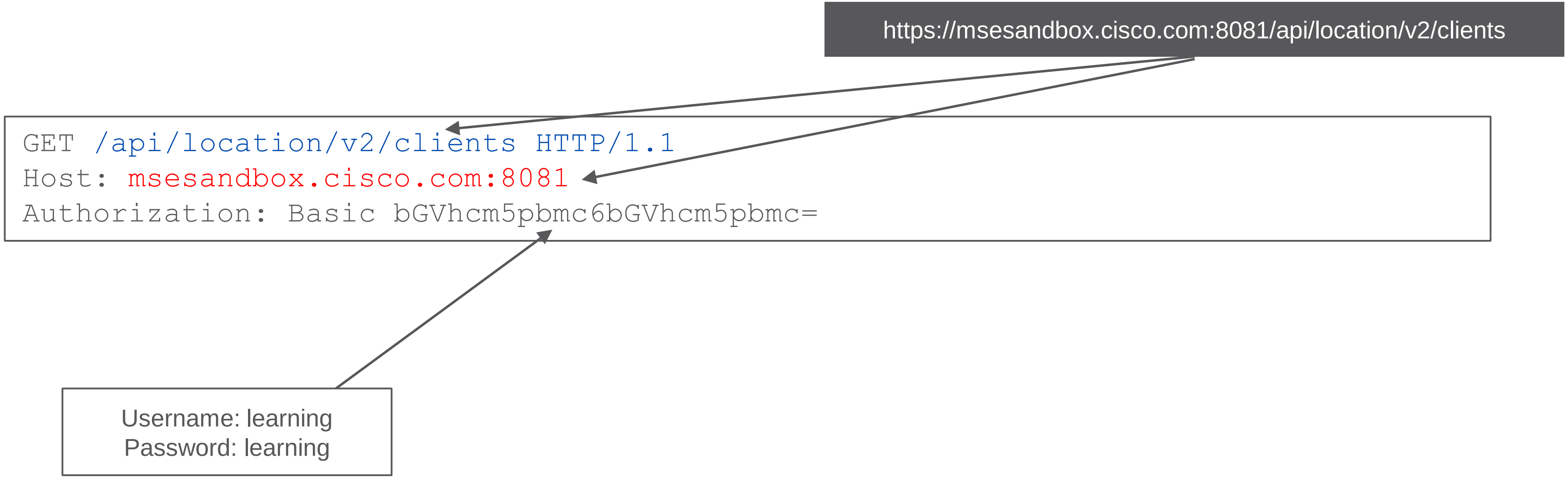
Access Details:
Endpoint URL: https://msesandbox.cisco.com:8081/api/location/v2/

Basic Authentication Credentials:

- Username: learning
- Password: learning

Additional Information:

“Get All Clients” Request

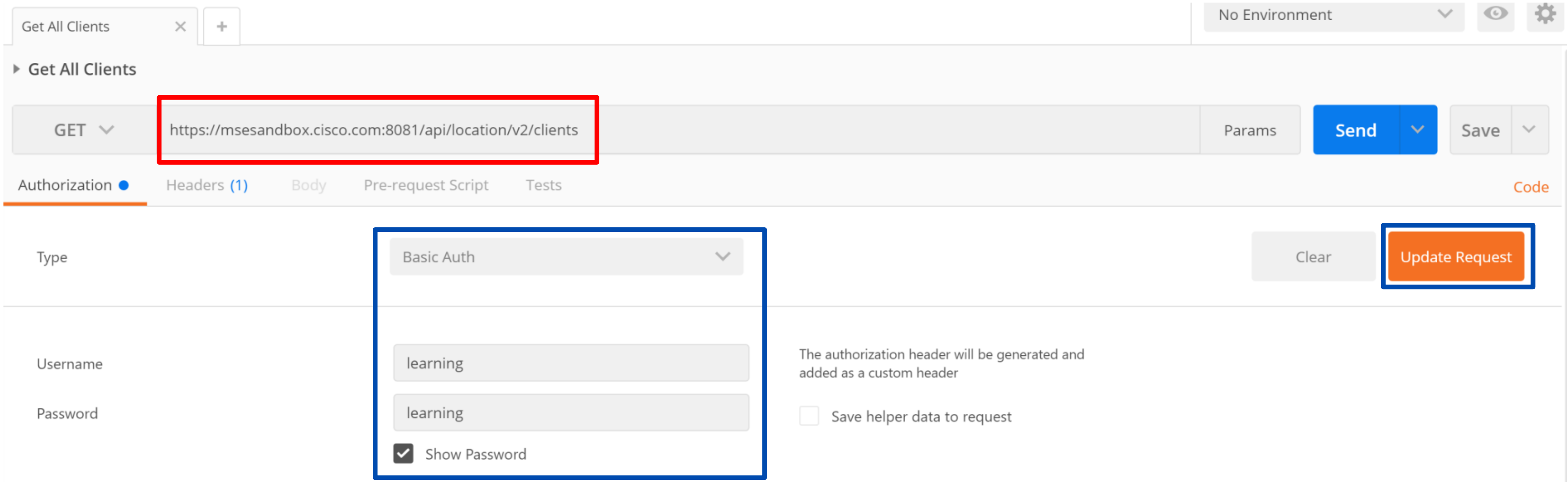


https://msesandbox.cisco.com:8081/api/location/v2/clients

```
GET /api/location/v2/clients HTTP/1.1
Host: msesandbox.cisco.com:8081
Authorization: Basic bGVhcm5pbmc6bGVhcm5pbmc=
```

Username: learning
Password: learning

Build Request in Postman Tool



GET https://msesandbox.cisco.com:8081/api/location/v2/clients

Type: Basic Auth

Username: learning

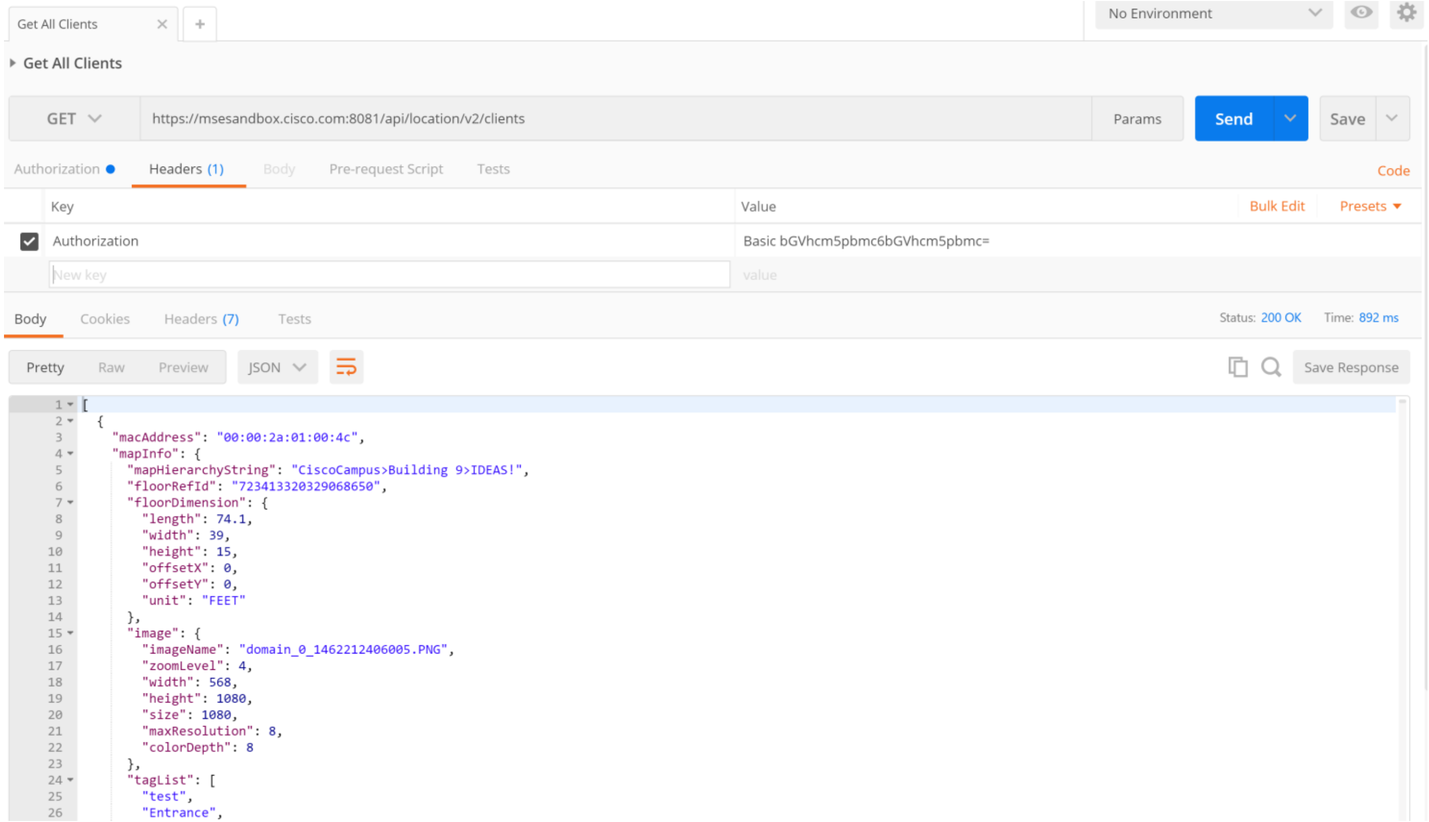
Password: learning

☑ Show Password

The authorization header will be generated and added as a custom header

□ Save helper data to request

Execute Request in Postman Tool



GET https://msesandbox.cisco.com:8081/api/location/v2/clients

	Key	Value
☑	Authorization	Basic bGVhcm5pbmc6bGVhcm5pbmc=
	New key	value

Status: 200 OK Time: 892 ms

```
[
  {
    "macAddress": "00:00:2a:01:00:4c",
    "mapInfo": {
      "mapHierarchyString": "CiscoCampus>Building 9>IDEAS!",
      "floorRefId": "723413320329068650",
      "floorDimension": {
        "length": 74.1,
        "width": 39,
        "height": 15,
        "offsetX": 0,
        "offsetY": 0,
        "unit": "FEET"
      },
      "image": {
        "imageName": "domain_0_1462212406005.PNG",
        "zoomLevel": 4,
        "width": 568,
        "height": 1080,
        "size": 1080,
        "maxResolution": 8,
        "colorDepth": 8
      },
      "tagList": [
        "test",
        "Entrance",
```

“Get All Clients” Response

Single Record

```
 {
    "macAddress": "00:00:2a:01:00:47",
    "mapInfo": {
      "mapHierarchyString": "CiscoCampus>Building 9>IDEAS!>CakeBread",
      "floorRefId": "723413320329068650",
...
      "image": {
        "imageName": "domain_0_1462212406005.PNG",
...
      },
... },
    "mapCoordinate": {
      "x": 21.153776,
      "y": 27.427116,
      "z": 0,
      "unit": "FEET"
    },

    "ipAddress": [
      "10.10.20.230"
    ],

    "band": "IEEE_802_11_B",
    "apMacAddress": "00:2b:01:00:0a:00",
    "dot11Status": "ASSOCIATED",
...
  },
```

Python: Execute Generic CMX Request

```
###################################
# DevNet CMX Lab
cmx_url_prefix = "https://msesandbox.cisco.com:8081/api/"
cmx_username = 'learning'
cmx_password = 'learning'
###################################
def CMX_GET(url_suffix):

    url = cmx_url_prefix + url_suffix

    header = {"Accept": 'application/json'}

    # To supress warning when using DevNet CMX Sandbox
    requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

    response = requests.get(url, headers=header, auth=HTTPBasicAuth(cmx_username,
cmx_password),verify=False)

    jsonObject = response.json()

    return jsonObject
```

Python: Build “Get All Clients” CMX Request

```
def CMX10_Get_All_Clients():

    url_suffix = "location/v2/clients"

    return CMX_GET(url_suffix)
```

Python: Print All Clients App

```
from CMX10_lib_dvulovic import CMX10_Get_All_Clients

json = CMX10_Get_All_Clients()

for client in json:

   print ('Client MAC {} (IP {}) status {} (AP {})'.
format(client['macAddress'],client['ipAddress'][0],client['dot11Status'],client['apMacAddress']))
```

Python: App Output (CMX Sandbox)

```
Client MAC 00:00:2a:01:00:4c (IP 10.10.20.235) status ASSOCIATED (AP 00:2b:01:00:0b:00)
Client MAC 00:00:2a:01:00:4b (IP 10.10.20.234) status ASSOCIATED (AP 00:2b:01:00:0b:00)
Client MAC 00:00:2a:01:00:4a (IP 10.10.20.233) status ASSOCIATED (AP 00:2b:01:00:0b:00)
Client MAC 00:00:2a:01:00:49 (IP 10.10.20.232) status ASSOCIATED (AP 00:2b:01:00:0b:00)
Client MAC 00:00:2a:01:00:48 (IP 10.10.20.231) status ASSOCIATED (AP 00:2b:01:00:0a:00)
Client MAC 00:00:2a:01:00:46 (IP 10.10.20.229) status ASSOCIATED (AP 00:2b:01:00:0a:00)
Client MAC 00:00:2a:01:00:44 (IP 10.10.20.227) status ASSOCIATED (AP 00:2b:01:00:0a:00)
Client MAC 00:00:2a:01:00:43 (IP 10.10.20.226) status ASSOCIATED (AP 00:2b:01:00:0a:00)
Client MAC 00:00:2a:01:00:42 (IP 10.10.20.225) status ASSOCIATED (AP 00:2b:01:00:0a:00)
Client MAC 00:00:2a:01:00:40 (IP 10.10.20.223) status ASSOCIATED (AP 00:2b:01:00:09:00)
Client MAC 00:00:2a:01:00:3f (IP 10.10.20.222) status ASSOCIATED (AP 00:2b:01:00:09:00)
Client MAC 00:00:2a:01:00:3e (IP 10.10.20.221) status ASSOCIATED (AP 00:2b:01:00:09:00)
Client MAC 00:00:2a:01:00:3c (IP 10.10.20.219) status ASSOCIATED (AP 00:2b:01:00:09:00)
Client MAC 00:00:2a:01:00:3b (IP 10.10.20.218) status ASSOCIATED (AP 00:2b:01:00:09:00)
Client MAC 00:00:2a:01:00:37 (IP 10.10.20.214) status ASSOCIATED (AP 00:2b:01:00:08:00)
Client MAC 00:00:2a:01:00:35 (IP 10.10.20.212) status ASSOCIATED (AP 00:2b:01:00:08:00)
Client MAC 00:00:2a:01:00:34 (IP 10.10.20.211) status ASSOCIATED (AP 00:2b:01:00:08:00)
Client MAC 00:00:2a:01:00:31 (IP 10.10.20.208) status ASSOCIATED (AP 00:2b:01:00:08:00)
Client MAC 00:00:2a:01:00:2e (IP 10.10.20.205) status ASSOCIATED (AP 00:2b:01:00:07:00)
Client MAC 00:00:2a:01:00:2d (IP 10.10.20.204) status ASSOCIATED (AP 00:2b:01:00:07:00)
…
Client MAC 00:00:2a:01:00:08 (IP 10.10.20.167) status ASSOCIATED (AP 00:2b:01:00:02:00)
Client MAC 00:00:2a:01:00:07 (IP 10.10.20.166) status ASSOCIATED (AP 00:2b:01:00:02:00)
Client MAC 00:00:2a:01:00:06 (IP 10.10.20.165) status ASSOCIATED (AP 00:2b:01:00:02:00)
```

“Get All APs” Request

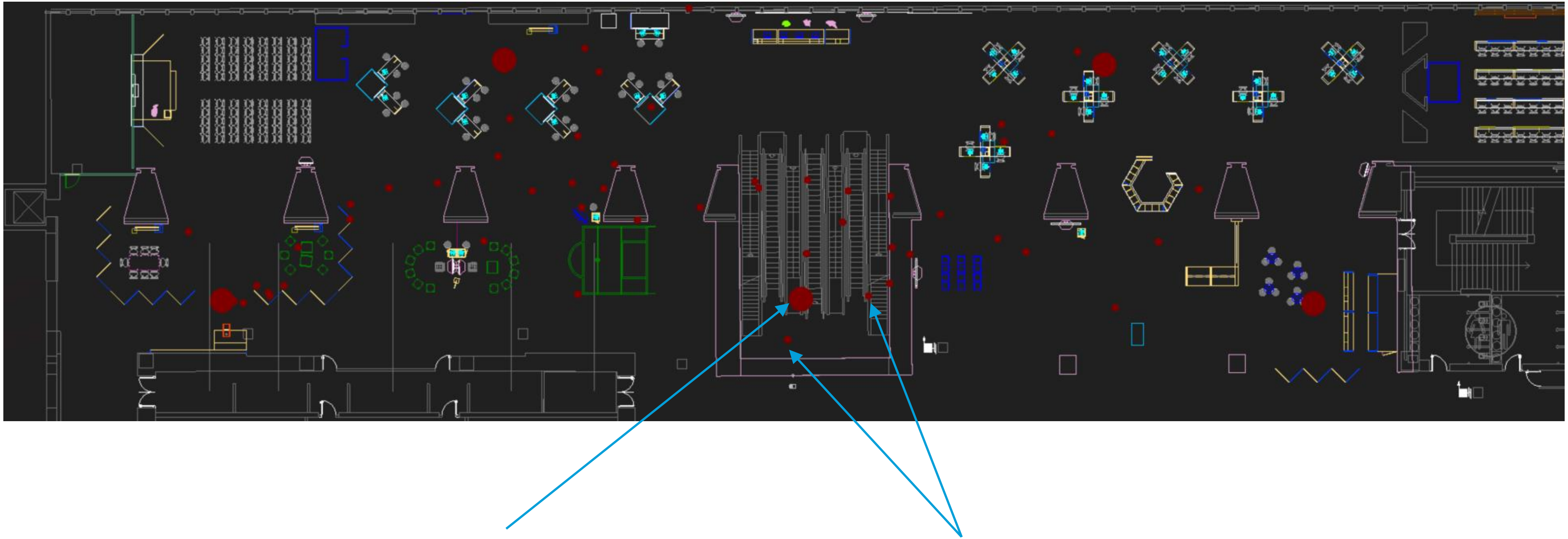
```
GET /api/config/v1/aps HTTP/1.1
Host: msesandbox.cisco.com:8081
Authorization: Basic bGVhcm5pbmc6bGVhcm5pbmc=
```

“Get All APs” Response

Single Record

```
 {
    "name": "T1-7",
    "radioMacAddress": "00:2b:01:00:08:00",
    "floorId": 723413320329068650
…
    "mapCoordinates": {
      "x": 3.5704226,
      "y": 69.43445,
      "z": 15,
      "unit": "FEET"
    },
    "apInterfaces": [
      {
        "band": "IEEE_802_11_B",
        "slotNumber": 0,
        "channelAssignment": 1,
        "channelNumber": 1,
        "txPowerLevel": 1,
        "txPowerControl": 1,
        "unit": "DEGREE",
…
      },
...
    ],
…
  },
```

Example App: FloorVisualizer



AP

Clients

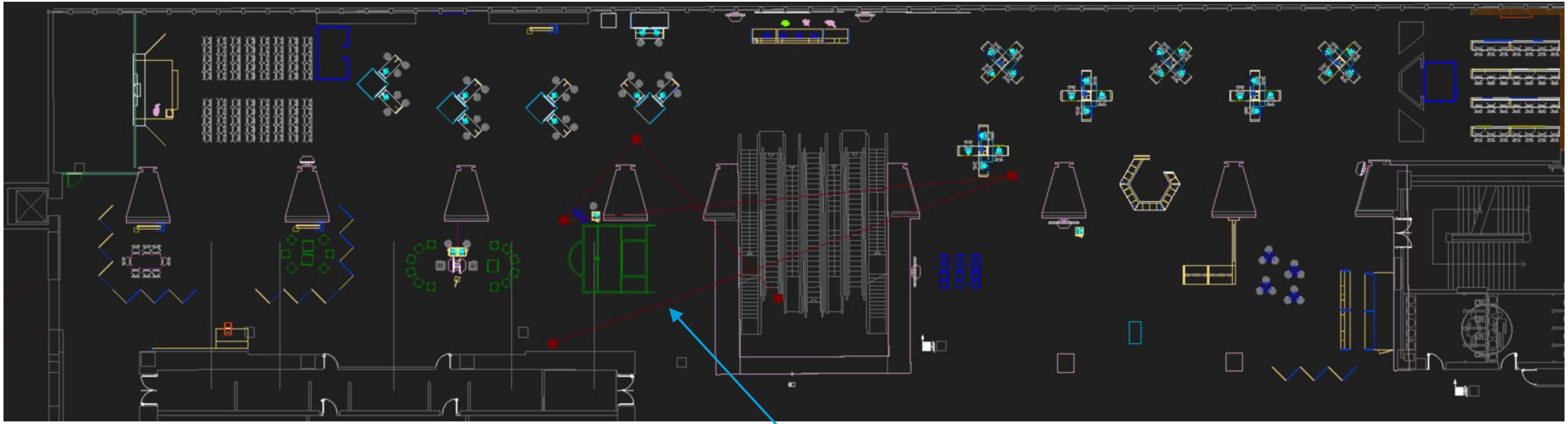
“Get Client History” Request

```
GET /api/location/v1/history/clients/00:00:2a:01:00:05 HTTP/1.1
Host: msesandbox.cisco.com:8081
Authorization: Basic bGVhcm5pbmc6bGVhcm5pbmc=
```

“Get Client History” Response

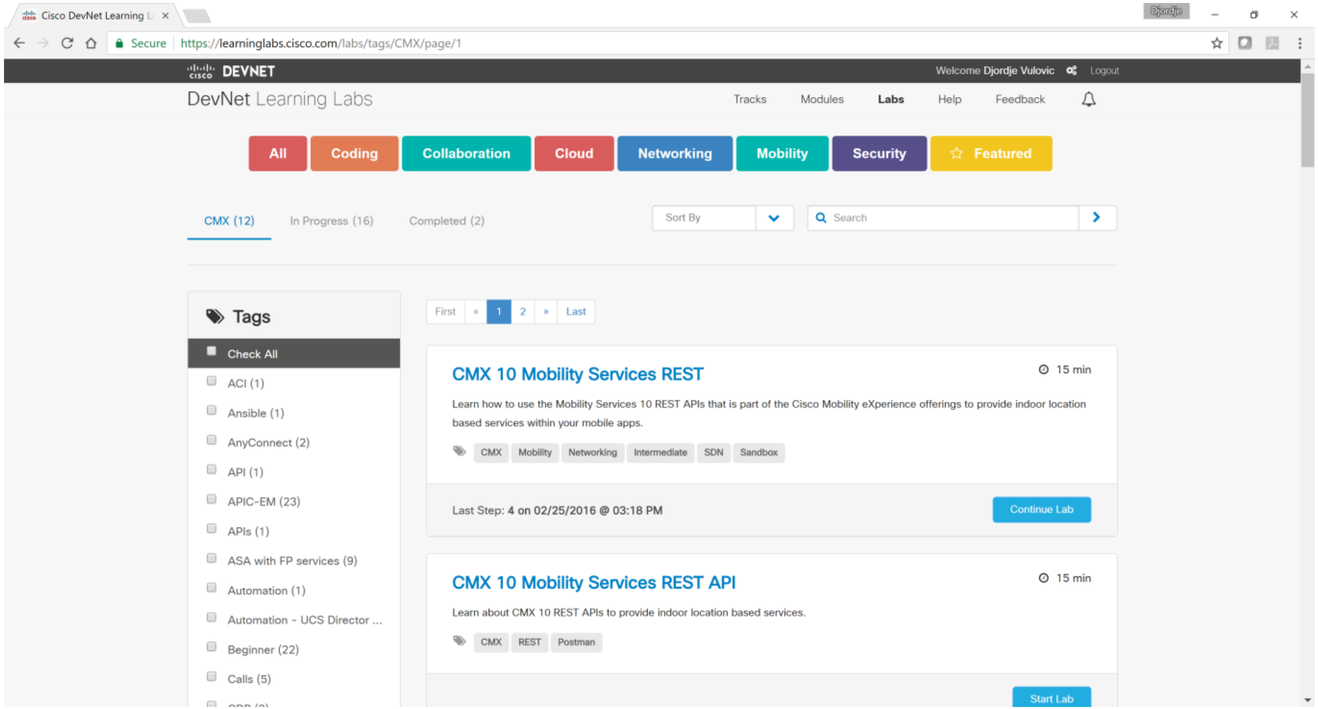
```
[
  {
    "macAddress": "00:00:2a:01:00:05",
    "mapCoordinate": {
      "x": 122.00833,
      "y": 28.161898,
      "z": 0,
      "unit": "FEET"
    },
    "statistics": {
      "currentServerTime": "2017-03-15T11:28:38.573+0000",
      "firstLocatedTime": "2017-03-14T15:54:53.224+0000",
      "lastLocatedTime": "2017-03-15T11:28:32.524+0000"
    },
  },
  {
    "macAddress": "00:00:2a:01:00:05",
    "mapCoordinate": {
      "x": 170.88286,
      "y": 11.773684,
      "z": 0,
      "unit": "FEET"
    },
    "statistics": {
      "currentServerTime": "2017-03-15T11:28:38.578+0000",
      "firstLocatedTime": "2017-03-14T15:54:53.224+0000",
      "lastLocatedTime": "2017-03-15T11:28:28.533+0000"
    },
  },
…
```

Example App: ClientTracker



Client Movement

CMX Learning Labs



CMX 10 Mobility Services REST 15 min

Learn how to use the Mobility Services 10 REST APIs that is part of the Cisco Mobility eXperience offerings to provide indoor location based services within your mobile apps.

CMX Mobility Networking Intermediate SDN Sandbox

Last Step: 4 on 02/25/2016 @ 03:18 PM

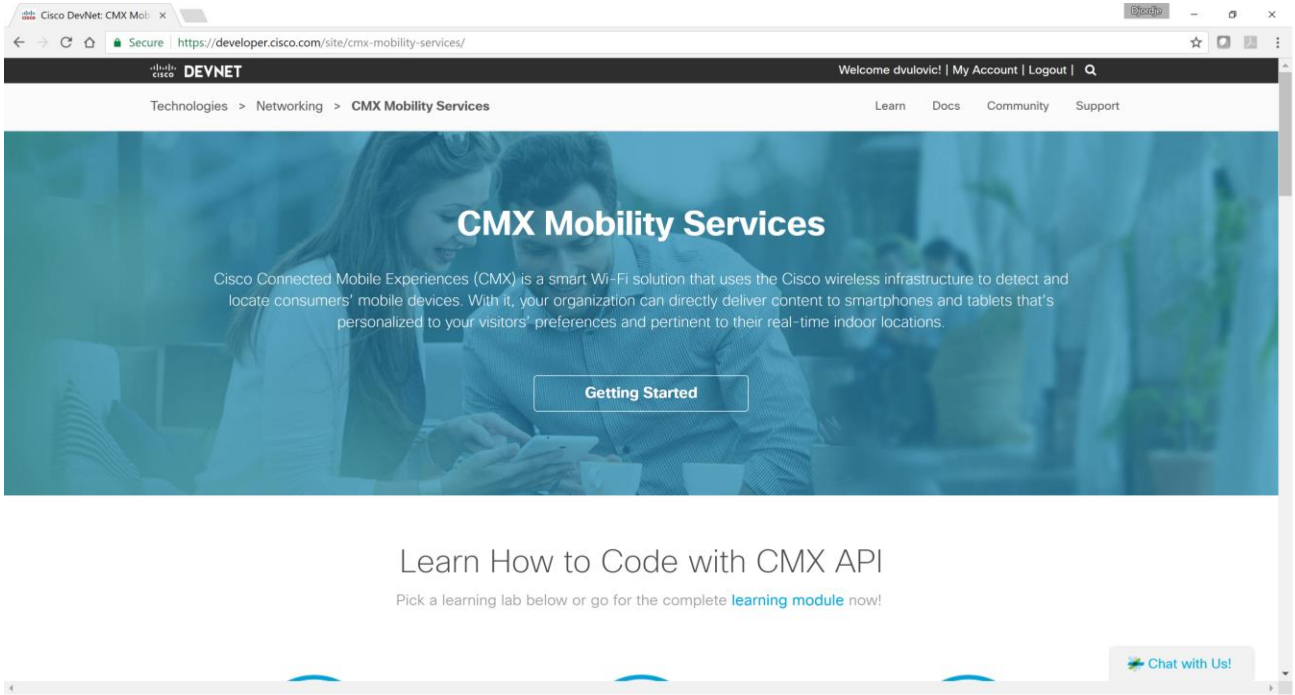
CMX 10 Mobility Services REST API 15 min

Learn about CMX 10 REST APIs to provide indoor location based services.

CMX REST Postman

https://learninglabs.cisco.com/labs/tags/CMX/page/1

DevNet CMX Page



Cisco DevNet: CMX Mob ×
Djordje
Secure
https://developer.cisco.com/site/cmx-mobility-services/
DEVNET
Welcome dvulovic! | My Account | Logout |
Technologies > Networking > CMX Mobility Services
Learn
Docs
Community
Support
Getting Started
Chat with Us!

CMX Mobility Services

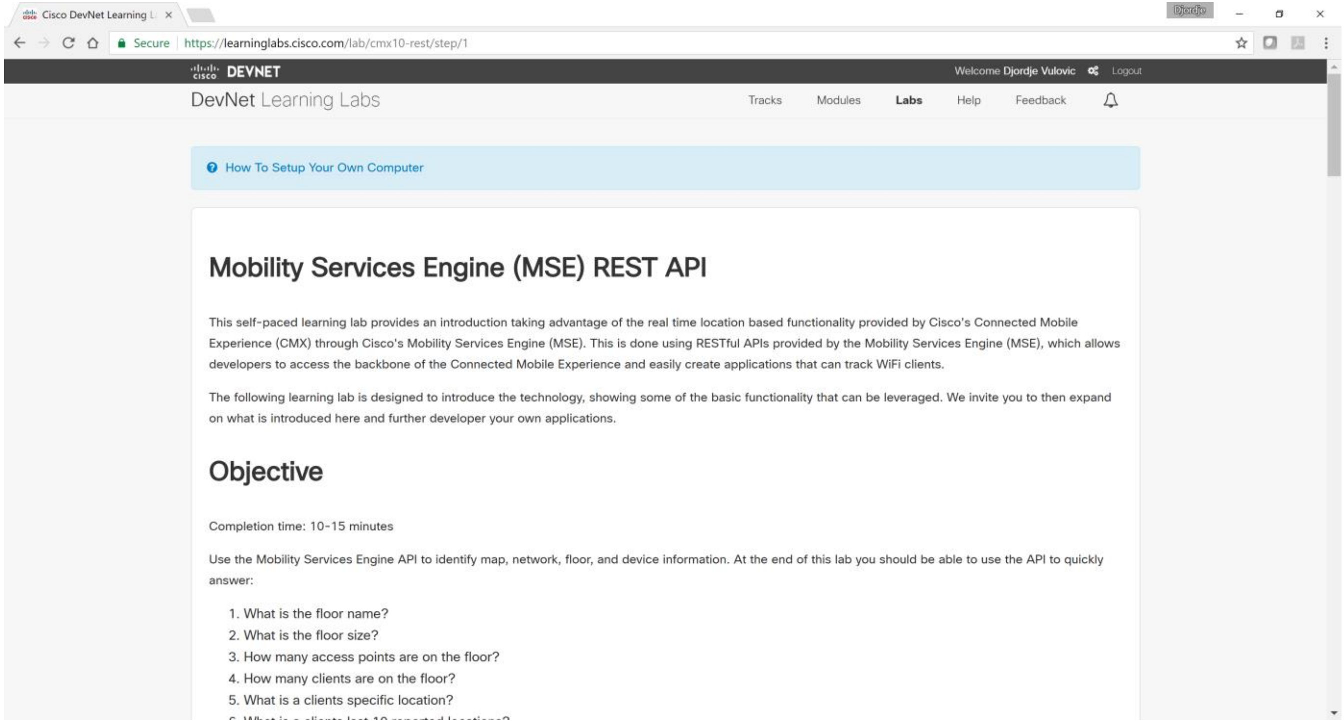
Cisco Connected Mobile Experiences (CMX) is a smart Wi-Fi solution that uses the Cisco wireless infrastructure to detect and locate consumers' mobile devices. With it, your organization can directly deliver content to smartphones and tablets that's personalized to your visitors' preferences and pertinent to their real-time indoor locations.

Learn How to Code with CMX API

Pick a learning lab below or go for the complete learning module now!

https://developer.cisco.com/site/cmx-mobility-services/

Mobility Services Engine (MSE) REST API



Mobility Services Engine (MSE) REST API

This self-paced learning lab provides an introduction taking advantage of the real time location based functionality provided by Cisco's Connected Mobile Experience (CMX) through Cisco's Mobility Services Engine (MSE). This is done using RESTful APIs provided by the Mobility Services Engine (MSE), which allows developers to access the backbone of the Connected Mobile Experience and easily create applications that can track WiFi clients.

The following learning lab is designed to introduce the technology, showing some of the basic functionality that can be leveraged. We invite you to then expand on what is introduced here and further developer your own applications.

Objective

Completion time: 10-15 minutes

Use the Mobility Services Engine API to identify map, network, floor, and device information. At the end of this lab you should be able to use the API to quickly answer:

1. What is the floor name?
2. What is the floor size?
3. How many access points are on the floor?
4. How many clients are on the floor?
5. What is a clients specific location?
6. What is a clients last 10 reported locations?

https://learninglabs.cisco.com/lab/cmx10-rest/step/1 cisco