Welcome to the 2nd session of the Cisco DevNet webinar series

- Use the Q and A panel to ask questions.
- Use the Chat panel to communicate with attendees and panelists.
- A link to a recording of the session will be sent to all registered attendees.
- Please take the feedback survey at the end of the webinar.
Joining You Today:

Matt Denapoli
Developer Evangelist
DevNet, Cisco

Giuseppe Cinque
Manager for the Emerging Technologies
NetAcad, Cisco
Next DevNet Session

THIRD SESSION:
Intent Networks

25 October – 9:00 A.M. PT

Intro to Coding
Giuseppe Cinque & Wadih Zaatar

Solutions & Marketing
September 2017
The world is becoming connected and programmable.
Introduction to IoT

Learn how the Internet of Things (IoT) and the digital transformation of business create new value and new job opportunities.

You can enroll today to learn more  http://bit.ly/IntroIoT
WE’RE ESSENTIALLY REDEFINING NETWORKING.
WHERE THE NETWORK IS PROGRAMMABLE
Module 01

Intro to Coding

Matthew DeNapoli
DevNet Developer Evangelist

Networking Academy | DEVNET
Agenda

- Getting Started
- Learning APIC-EM
- APIC-EM with Postman – HTTP Calls & Generate Code
- Calling APIC-EM REST APIs with Python
- Q&A
APIC-EM Applications and Use Cases

• Easy QoS
  • Application Priority

• Plug-n-Play
  • Agent based control

• IWAN
  • Policy based automated deployment

• Path Trace
  • Path troubleshooting
APIC-EM Uses REST

Cisco APIC-EM REST APIs

- Hosts
- Devices
- Users
- + more

How does this work?
Anatomy of a REST Request

**Method**
- POST, GET, PUT, DELETE (CRUD)

**URL**
- Example: http://{APIC-EMController}/api/v1/host

**Authentication**
- Basic HTTP, OAuth, none, Custom

**Custom Headers**
- HTTP Headers
- Example: Content-Type: application/json

**Request Body**
- JSON or XML containing data needed to complete request
APIC-EM Example: Post Ticket

Application Policy Infrastructure Controller (APIC) Enterprise Module (EM)

POST http://{APIC-EMController}/api/v1/ticket

Request

Authorization Code returned in JSON

Response

3rd Party App
APIC-EM Example: Get Host

Application Policy Infrastructure Controller (APIC) Enterprise Module (EM)

GET http://{APIC-EMController}/api/v1/host

Request

List of Hosts returned in JSON

Response

3rd Party App
Using the API Reference Documentation

<table>
<thead>
<tr>
<th>Available APIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
</tr>
<tr>
<td>Flow Analysis</td>
</tr>
<tr>
<td>IP Geolocation</td>
</tr>
<tr>
<td>IP Pool Manager</td>
</tr>
<tr>
<td>Inventory</td>
</tr>
<tr>
<td>Network Discovery</td>
</tr>
<tr>
<td>Network Plug and Play</td>
</tr>
<tr>
<td>PKI Broker Service</td>
</tr>
<tr>
<td>Policy Administration</td>
</tr>
<tr>
<td>Role Based Access Control</td>
</tr>
<tr>
<td>Scheduler</td>
</tr>
<tr>
<td>Task</td>
</tr>
<tr>
<td>Topology</td>
</tr>
</tbody>
</table>

## Inventory

APIC-EM Service API based on the Swagger™ 1.2 specification

[Terms of Service](#)

### device-credential : Device Credential API

<table>
<thead>
<tr>
<th>Operations</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show/Hide</td>
<td>List Operations</td>
</tr>
</tbody>
</table>

### discovery : Discovery API

<table>
<thead>
<tr>
<th>Operations</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show/Hide</td>
<td>List Operations</td>
</tr>
</tbody>
</table>

### host : host API

<table>
<thead>
<tr>
<th>Method</th>
<th>Route</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/host</td>
<td>Retrieve hosts</td>
</tr>
<tr>
<td>GET</td>
<td>/host/count</td>
<td>Gives total number of hosts</td>
</tr>
<tr>
<td>GET</td>
<td>/host/{id}</td>
<td>Retrieves host based on id</td>
</tr>
</tbody>
</table>

### interface : Interface API

<table>
<thead>
<tr>
<th>Operations</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show/Hide</td>
<td>List Operations</td>
</tr>
</tbody>
</table>

### location : Location API

<table>
<thead>
<tr>
<th>Operations</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show/Hide</td>
<td>List Operations</td>
</tr>
</tbody>
</table>

### network-device : network-device API

<table>
<thead>
<tr>
<th>Operations</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show/Hide</td>
<td>List Operations</td>
</tr>
</tbody>
</table>

### network-device-config : Network Device Configuration API

<table>
<thead>
<tr>
<th>Operations</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show/Hide</td>
<td>List Operations</td>
</tr>
</tbody>
</table>

### tag : Tag API

<table>
<thead>
<tr>
<th>Operations</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show/Hide</td>
<td>List Operations</td>
</tr>
</tbody>
</table>
API Reference Guide Details

GET /host

Implementation Notes
Get Hosts

Response Class
Model: Model Schema

HostListResult {
  version (string, optional),
  response (array[HostDTO], optional)
}

HostDTO {
  hostName (string, optional): Name of the host,
  source (string): Source from which the host gets collected. Available option: 200 for inventory collection and 300 for trap based data collection,
  lastUpdated (string): Time when the host into last got updated,
  vitnid (string, optional): Vlan id of the host,
  connectedAPMacAddress (string, optional): Mac address of the AP to which wireless host gets connected,
  connectedAPName (string, optional): Name of the AP to which wireless host gets connected,
  connectedInterfaceId (string, optional): Id of the interface to which host gets connected,
  connectedInterfaceName (string, optional): Name of the interface to which host gets connected,
  connectedNetworkDeviceId (string): Id of the network device to which host gets connected,
  connectedNetworkDeviceIpAddress (string): Ip address of the network device to which host gets connected,
  hostIp (string): Ip address of the host,
}

Response Content Type: application/json

Parameters
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
<th>Parameter Type</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>limit</td>
<td></td>
<td>limit</td>
<td>query</td>
<td>string</td>
</tr>
<tr>
<td>offset</td>
<td></td>
<td>offset</td>
<td>query</td>
<td>string</td>
</tr>
<tr>
<td>sortBy</td>
<td></td>
<td>sortBy</td>
<td>query</td>
<td>string</td>
</tr>
</tbody>
</table>
APIC-EM with Postman
https://www.getpostman.com

Modern software is built on APIs.
Postman helps you develop APIs faster.

Chrome App  Mac App

Go ahead and download our apps, they're free!
Postman
Create a ticket

- **Method**: POST
- **URL**: https://198.18.129.100/api/v1/ticket

**Request body**:

```
{
  "username": "admin",
  "password": "C1sco12345"
}
```
Header Specification

<table>
<thead>
<tr>
<th>Authorization</th>
<th>Headers (1)</th>
<th>Body</th>
<th>Pre-request Script</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Content-Type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>application/json</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>value</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Header Key**

**Header Value**
Ticket Returned in Response Body

```
{
  "response": {
    "serviceTicket": "ST-7-YSSUUXyh22mtKbE2J63m-cas",
    "idleTimeout": 1800,
    "sessionTimeout": 21600
  },
  "version": "1.0"
}
```
Use ticket in Header in all API Calls

**GET**

https://198.18.129.100/api/v1/host

<table>
<thead>
<tr>
<th>Authorization</th>
<th>Headers (1)</th>
<th>Body</th>
<th>Pre-request Script</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X-Auth-Token</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>value</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Header**

**Ticket**
### APIC-EM – List of Hosts

#### URL
- **Method**: GET
- **URL**: `https://198.18.129.100/api/v1/host`

#### Header
- **Authorization**: X-AUTH-TOKEN
- **X-Auth-Token**: ST-7-YSSUUXYh22mKbE2J63m-cas

#### Body
- **HTTP Response**: 200 OK
- **Time**: 185 ms

#### Response
```json
{   "response": [       {           "id": "4c60d6a7-4812-40d6-a337-773af2625e56",           "hostIp": "65.1.1.86",           "hostMac": "00:24:7d:43:59:d8",           "hostType": "wireless",           "connectedNetworkDeviceId": "17184480-2617-42c3-b267-4f3ade5f794a9",           "connectedNetworkDeviceIpAddress": "55.1.1.3",           "connectedAPMacAddress": "68:bc:0c:63:4a:be0",           "connectedAPName": "AP7081.059f.19ca",           "vlanId": "600",           "lastUpdated": "1467837609856",           "avgUpdateFrequency": "1800",           "source": "300",           "pointOfPresence": "5a3db62-5def-40a1-be98-944ba2a7d863",           "pointOfAttachment": "5a3db62-5def-40a1-be98-944ba2a7d863"       }   ]
```
Using Postman to Generate Code

POST https://198.18.129.100/api/v1/ticket

Authorization Headers (2) Body Pre-request Script Tests

- form-data
- x-www-form-urlencoded
- raw
- binary

JSON (application/json)

```
{
    "username": "admin",
    "password": "Cisco12345"
}
```

Body Cookies Headers (9) Tests

Pretty Raw Preview JSON

```
{
    "response": {
        "serviceTicket": "ST-4363-MBfA2YQc7NlELXJRY-cas",
        "idleTimeout": 1800,
        "sessionTimeout": 21600
    },
    "version": "1.0"
}
```
Selecting the Code to Generate

---

**Python Requests**

- **HTTP**
- C (LibCurl)
- cURL
- C# (RestSharp)
- Go
- Java
- JavaScript
- NodeJS
- Objective-C (NSURL)
- OCaml (Cohttp)
- PHP
- **Python**
- Ruby (NET::Http)
- Shell
- Swift (NSURL)

```python
import requests

url = "sandboxapic.cisco.com/api/v1/ticket"
headers = {
    "Content-Type": "application/json",
    "Accept": "application/json",
    "Authorization": "Bearer abc123",
    "fn": "user1",
    "role": "admin",
    "return_ssl_verification": False
}
response = requests.request("POST", url, headers=headers)
```

---

**Copy to Clipboard**
import requests

url = "https://198.18.129.100/api/v1/ticket"

payload = "{"username":"admin","password":"C1sco12345"}"

headers = {
    'content': "application/json",
    'content-type': "application/json",
    'cache-control': "no-cache",
    'postman-token': "90f57fc2-ade2-2884-c133-ec6aa2f479f3"
}

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
REST Demo – Using Postman

- **Get Hosts**
  - Method: GET
  - Headers: ‘X-Auth-Token’ (insert your ticket value)
  - URL: http://<APIC-EMController>/api/v1/host

- **Get Devices**
  - Method: GET
  - Headers: ‘X-Auth-Token’ (insert your ticket value)
  - URL: http://<APIC-EMController>/api/v1/network-device

- **Get Users**
  - Method: GET
  - Headers: ‘X-Auth-Token’ (insert your ticket value)
  - URL: http://<APIC-EMController>/api/v1/user
APIC-EM with Python
import requests
import json

url = 'https://198.18.129.100/api/v1/ticket'
payload = {"username":"admin","password":"C1sco12345"}
header = {"content-type": "application/json"}
response= requests.post(url,data=json.dumps(payload), headers=header, verify=False)

print(response.text)
Getting Ticket Function

apic_em_ip = "https://198.18.129.100/api/v1"

def get_token(url):
    api_call = "/ticket"
    payload = {"username": "admin", "password": "C1sco12345"}
    headers = {"content-type": "application/json"}
    url += api_call
    response = requests.post(url, data=json.dumps(payload),
                              headers=headers, verify=False).json()

    return response["response"]['serviceTicket']
def get_device_id(token, url):
    api_call = "/network-device"
    headers = {
        "X-AUTH-TOKEN": token
    }
    url += api_call
    response = requests.get(url, headers=headers, verify=False).json()
    for item in response['response']:
        if item['role'] == 'ACCESS':
            return item['id']
This is the Digital Transformation

Introduction to IoT

Learn how the Internet of Things (IoT) and the digital transformation of business create new value and new job opportunities.

Next DevNet Session

THIRD SESSION:

Intent Networks

25 October – 9:00 A.M. PT
