



Cisco DevNet Series

Session 2: Introduction to Coding

Speaker: Matt Denapoli

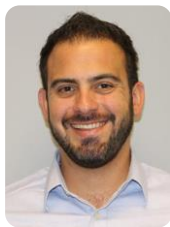
Hostess: Kara Sullivan | Cisco Networking Academy

14 September 2017

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:

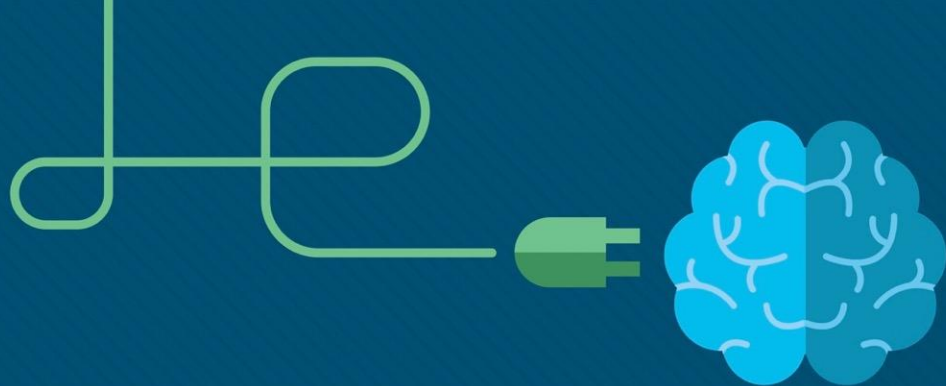
Intent Networks



25 October – 9:00 A.M. PT

Register at: <http://bit.ly/DevNetSession3>





Intro to Coding

Giuseppe Cinque & Wadih Zaatar

Solutions & Marketing
September 2017





The world is becoming connected and programmable



This is the Digital Transformation



Cisco Networking Academy

Courses ▾

Careers ▾

Get Started ▾

About Us ▾

Log In English ▾

🔍 Search

Home / Courses / Introduction to IoT

Introduction to IoT

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

Enroll Now



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 | 

Agenda

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

Learning APIC-EM



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



Easy to use:

- In mobile apps
- In console apps
- In web apps



Cisco APIC-EM REST APIs

- Hosts
- Devices
- Users
- + more

How does this work?

```
apic-em-examples — bash — 90x20
Hosts=
192.168.68.130
26.6.6.9
26.6.6.11
26.6.6.10
25.5.5.56
14.4.4.12
14.4.4.17
14.4.4.11
14.4.4.10
12.2.2.11
12.2.2.10
12.2.2.12
Policies=
419abccf-c2c9-4421-a96a-f3de3981ce5f
100c8e40-fa50-4faf-a099-c0b0436329c0
2a9460ba-7db2-4790-b1b3-8d13166c69e5
fdbccc11-ca01-4d5f-8bdc-636ecbb6f8b3
2c27bdea-ddef-4b77-ada6-a378d392bbe3
```



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

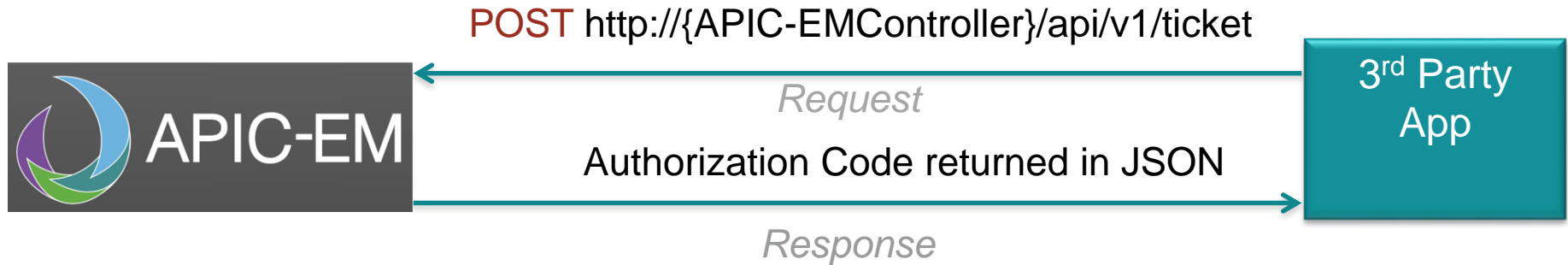
- HTTP Headers
- Example: `Content-Type: application/json`

Request Body

- JSON or XML containing data needed to complete request

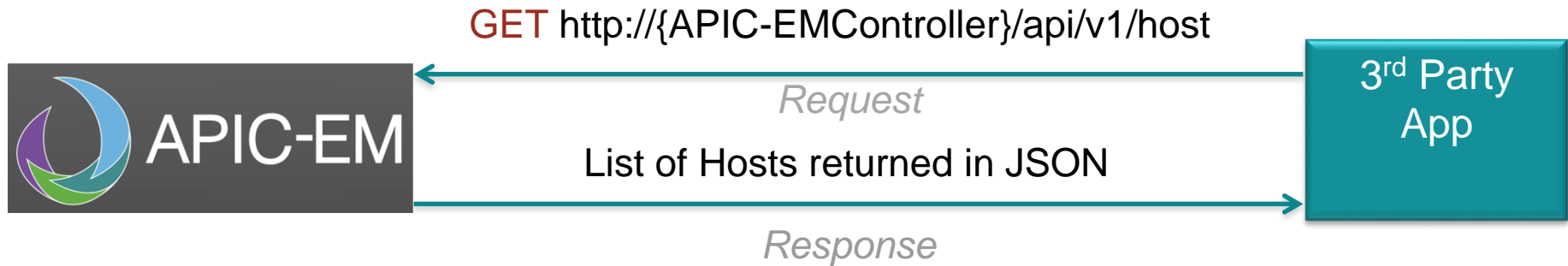
APIC-EM Example: Post Ticket

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







APIC-EM Example: Get Host

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



Using the API Reference Documentation

 APIC - Enterprise Module API   admin 

Available APIs

- [File](#)
- [Flow Analysis](#)
- [IP Geolocation](#)
- [IP Pool Manager](#)
- [Inventory](#)**
- [Network Discovery](#)
- [Network Plug and Play](#)
- [PKI Broker Service](#)
- [Policy Administration](#)
- [Role Based Access Control](#)
- [Scheduler](#)
- [Task](#)
- [Topology](#)

Inventory

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

[Terms of service](#)
[Cisco DevNet](#)

device-credential : Device Credential API [Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

discovery : Discovery API [Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

host : host API [Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

GET	/host	Retrieve hosts
GET	/host/count	Gives total number of hosts
GET	/host/{id}	Retrieves host based on id

interface : Interface API [Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

location : Location API [Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

network-device : network-device API [Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

network-device-config : Network Device Configuration API [Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

tag : Tag API [Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

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 info last got updated,
  vianId (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

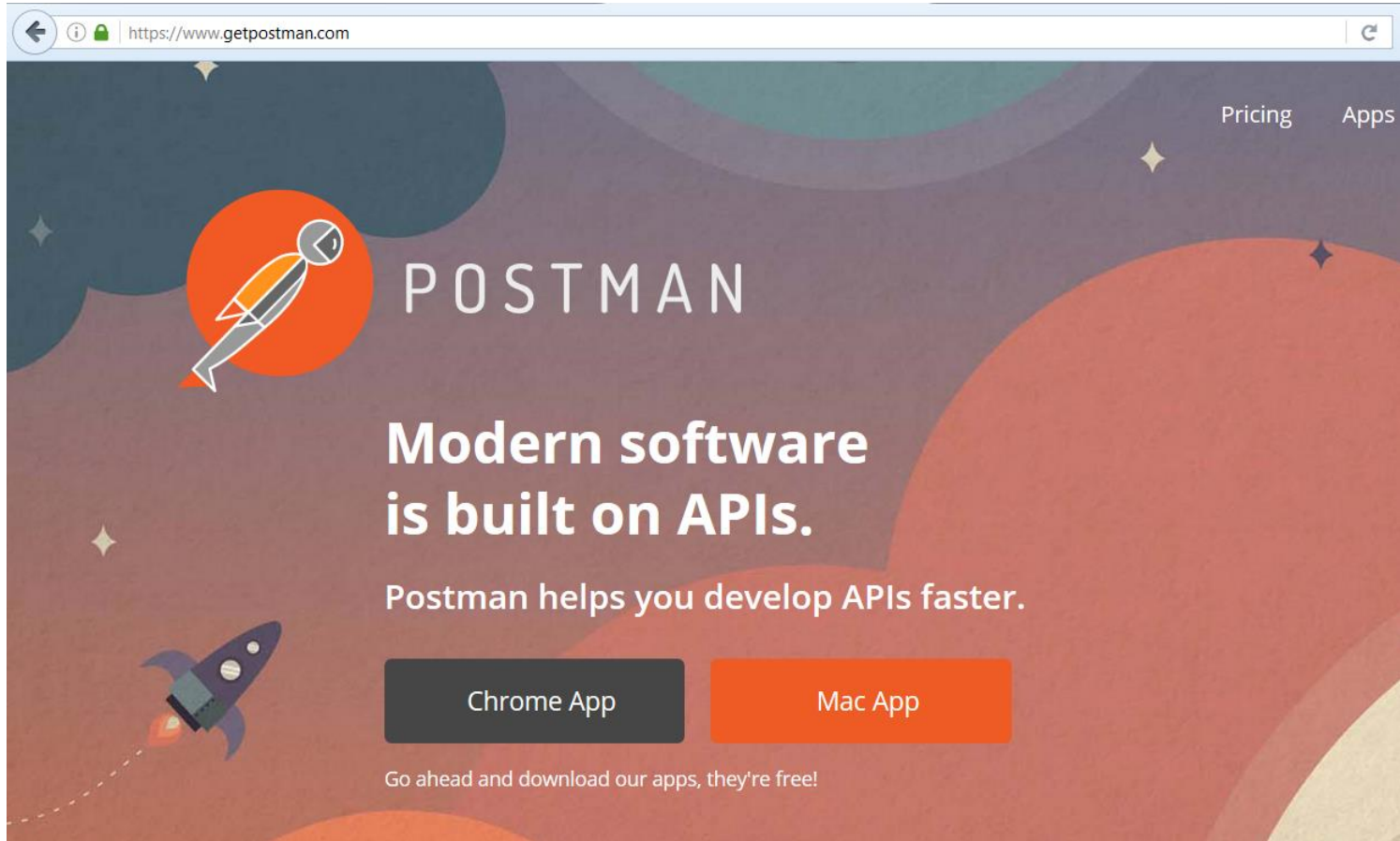
Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	limit	query	string
offset	<input type="text"/>	offset	query	string
sortBy	<input type="text"/>	sortBy	query	string

APIC-EM with Postman



<https://www.getpostman.com>



The screenshot shows the homepage of the Postman website. The browser's address bar at the top displays "https://www.getpostman.com". The page features a dark, space-themed background with stylized planets and stars. On the left, there is a large orange circle containing a white rocket icon. The word "POSTMAN" is written in large, white, uppercase letters. Below this, the main headline reads "Modern software is built on APIs." followed by the sub-headline "Postman helps you develop APIs faster." At the bottom, there are two buttons: a dark grey "Chrome App" button and an orange "Mac App" button. Below the buttons, a line of text says "Go ahead and download our apps, they're free!". In the top right corner, there are links for "Pricing" and "Apps".

https://www.getpostman.com

Pricing Apps

POSTMAN

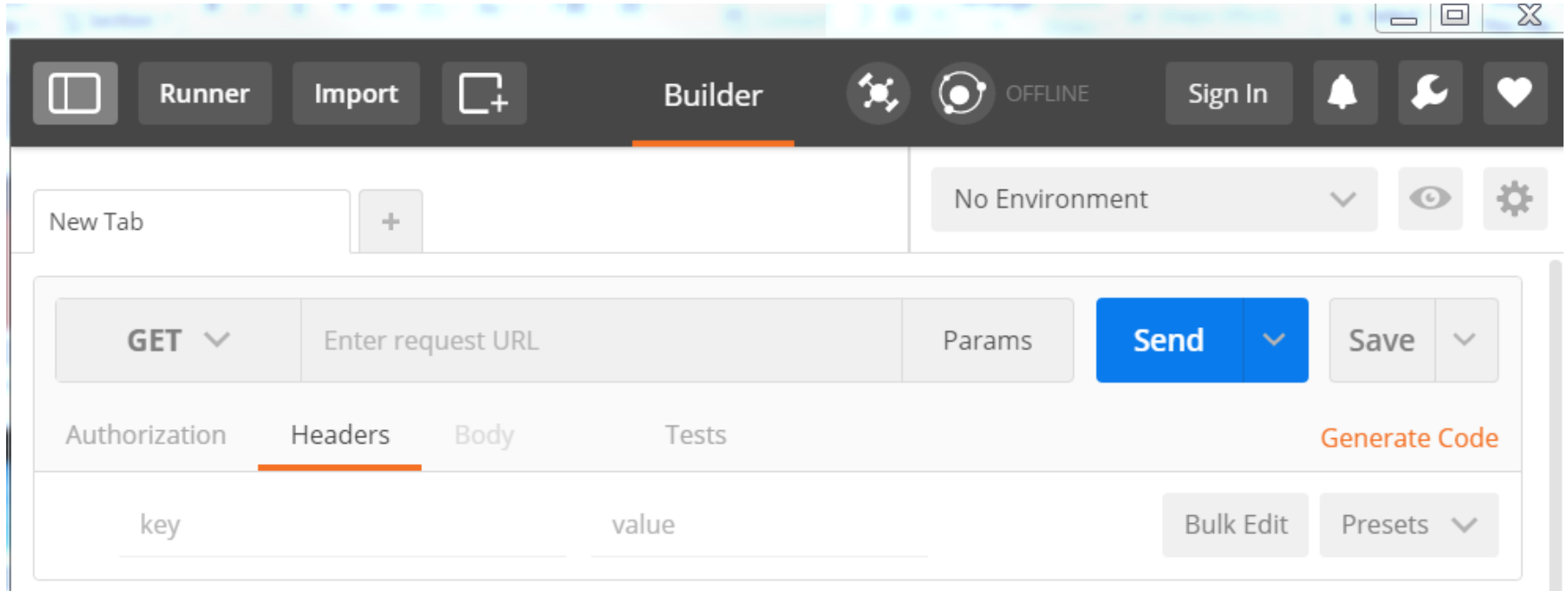
Modern software
is built on APIs.

Postman helps you develop APIs faster.

Chrome App Mac App

Go ahead and download our apps, they're free!

Postman



Create a ticket

The screenshot displays a REST client interface for configuring a POST request. At the top, the method is set to **POST** (indicated by an arrow labeled "method") and the URL is `https://198.18.129.100/api/v1/ticket` (indicated by an arrow labeled "url"). A **Send** button is visible on the right. Below the URL bar, the **Body** tab is selected. The content type is set to **JSON (application/json)**. The request body is defined as:

```
1 { "username": "admin",  
2   "password": "Cisco12345"  
3 }
```

An arrow labeled "Request body" points to the JSON content.

Header Specification

POST https://198.18.129.100/api/v1/ticket Params **Send**

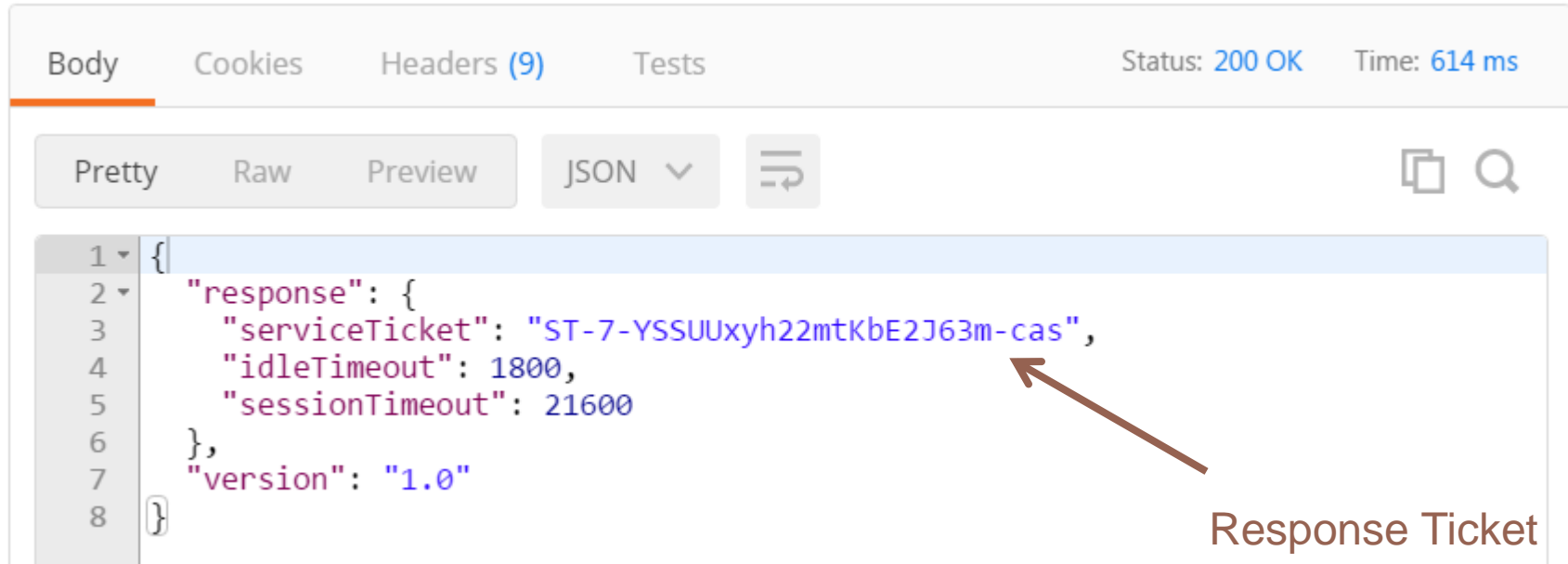
Authorization Headers (1) Body ● Pre-request Script Tests

<input checked="" type="checkbox"/>	Content-Type	application/json
	key	value

Header Key

Header Value

Ticket Returned in Response Body



The screenshot shows the 'Body' tab of a web browser's developer tools. The status bar indicates 'Status: 200 OK' and 'Time: 614 ms'. The response body is displayed in JSON format, showing a 'response' object with the following properties:

```
1 {  
2   "response": {  
3     "serviceTicket": "ST-7-YSSUxyh22mtKbE2J63m-cas",  
4     "idleTimeout": 1800,  
5     "sessionTimeout": 21600  
6   },  
7   "version": "1.0"  
8 }
```

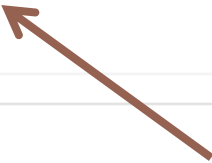
An arrow points to the value of the 'serviceTicket' property, which is 'ST-7-YSSUxyh22mtKbE2J63m-cas'.

Response Ticket
(save this)

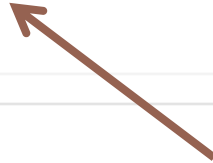
Use ticket in Header in all API Calls

The screenshot shows an API client interface with the following components:

- Method: GET
- URL: https://198.18.129.100/api/v1/host
- Params: (empty)
- Send button: (blue)
- Tab: Headers (1) (underlined in orange)
- Header list:
 - X-Auth-Token (checked) with value ST-7-YSSUUxyh22mtKbE2J63m-cas
- Input fields for key and value are visible below the header list.



Header



Ticket

APIC-EM – List of Hosts

The screenshot displays a REST client interface with the following components:

- Method:** GET (indicated by an arrow from the label "Method").
- URL:** https://198.18.129.100/api/v1/host (indicated by an arrow from the label "URL").
- Header:** X-Auth-Token: ST-7-YSSUUxyh22mtKbE2J63m-cas (indicated by an arrow from the label "Header").
- Status:** 200 OK (indicated by an arrow from the label "Return Code").
- Response Body:** A JSON object containing a list of host information (indicated by an arrow from the label "Response Body").

```
1 {
2   "response": [
3     {
4       "id": "4c60d6a7-4812-40d6-a337-773af2625e56",
5       "hostIp": "65.1.1.86",
6       "hostMac": "00:24:d7:43:59:d8",
7       "hostType": "wireless",
8       "connectedNetworkDeviceId": "17184480-2617-42c3-b267-4fade5f794a9",
9       "connectedNetworkDeviceIpAddress": "55.1.1.3",
10      "connectedAPMacAddress": "68:bc:0c:63:4a:b0",
11      "connectedAPName": "AP7081.059f.19ca",
12      "vlanId": "600",
13      "lastUpdated": "1467837609856",
14      "avgUpdateFrequency": "1800",
15      "source": "300",
16      "pointOfPresence": "5a3bdb62-5def-40a1-be98-944ba2a7d863",
17      "pointOfAttachment": "5a3bdb62-5def-40a1-be98-944ba2a7d863"
18    }
19  ]
}
```

Using Postman to Generate Code

The screenshot displays the Postman interface for a REST client. At the top, the request method is set to **POST** and the URL is `https://198.18.129.100/api/v1/ticket`. The **Code** tab is highlighted with a red box. The request body is set to **raw** with the content type **JSON (application/json)**. The request body contains the following JSON:

```
1 {  
2   "username": "admin",  
3   "password": "Cisco12345"  
}
```

The response section shows a **Status: 200 OK** and **Time: 476 ms**. The response body is displayed in the **JSON** view, showing the following JSON:

```
1 {  
2   "response": {  
3     "serviceTicket": "ST-4363-MBJA2YQEcan7LnELXJRY-cas",  
4     "idleTimeout": 1800,  
5     "sessionTimeout": 21600  
6   },  
7   "version": "1.0"  
8 }
```

The Cisco logo is visible in the bottom left corner.

Selecting the Code to Generate

The screenshot shows a web interface titled "GENERATE CODE SNIPPETS" with a close button in the top right corner. Below the title bar, there is a dropdown menu currently set to "Python Requests". To the right of this menu is an orange button labeled "Copy to Clipboard". The main area of the interface displays a code snippet for a REST client request. A dropdown menu is open, listing various programming languages and frameworks. The "Python" option is highlighted with a red box, and its sub-menu is also open, with "Requests" highlighted by another red box. The code snippet in the background is as follows:

```
sandboxapic.cisco.com/api/v1/ticket"
ername\":"devnetuser\","\n\password\":"Cisco123!\n}"
plication/json",
pe': "application/json",
rol': "no-cache",
ken': "c2392956-7a14-4b88-0378-588aa560a5f5"
ests.request("POST", url, data=payload, headers=headers)
text)
```

Generated Python Requests Code

GENERATE CODE SNIPPETS



Python Requests ▾

Copy to Clipboard

```
1 import requests
2
3 url = "https://198.18.129.100/api/v1/ticket"
4
5 payload = "{\"username\":\"admin\",\\n\"password\":\"C1sco12345\"\\n}"
6 headers = {
7     'cont': "application/json",
8     'content-type': "application/json",
9     'cache-control': "no-cache",
10    'postman-token': "90f57fc2-ade2-2884-c133-ec6aa2f479f3"
11 }
12
13 response = requests.request("POST", url, data=payload, headers=headers)
14
15 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



First APIC-EM REST call from 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"]
```

Getting Network Device ID/Config Functions

```
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']
```

Q&A



This is the Digital Transformation



Cisco Networking Academy

Courses ▾

Careers ▾

Get Started ▾

About Us ▾

Log In English ▾

🔍 Search

Home / Courses / Introduction to IoT

Introduction to IoT

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

Enroll Now



You can enroll today to learn more <http://bit.ly/IntroIoT>



Next DevNet Session

THIRD SESSION:

Intent Networks



25 October – 9:00 A.M. PT

Register at: <http://bit.ly/DevNetSession3>



