



APIC-EM

Concurso de Contenido Técnico- Julio & Agosto 2020

Edwin Portillo



Es un modulo empresarial del controlador de infraestructura de políticas de aplicaciones de Cisco que funciona como un controlador SDN de red empresarial. Puede ser un dispositivo solo de software virtual o un dispositivo físico, creando redes inteligentes abiertas y programables con API además transforma las políticas de intención empresarial en una configuración de red dinámica proporcionando un punto único para la automatización y control de toda la red.

Inicie sesión en APIC-EM



 CISCO

 APIC-EM

Cisco Application Policy Infrastructure Controller Enterprise Module

Username

Password

Log In

© 2009-2016 Cisco Systems, Inc., Cisco, Cisco Systems, and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries. The copyrights to certain works contained in this software are owned by other third parties and used and distributed under license. Certain components of this software are licensed under the GNU [GPL 2.0](#), [GPL 3.0](#), [LGPL 2.1](#), [LGPL 3.0](#) and [AGPL 3.0](#).

URL: <https://sandboxdnac.cisco.com/>

Usuario: devnetuser

Contraseña: Cisco123!

Página principal de APIC-EM

Documentación de API



Servicios

Aplicaciones

APIC - Enterprise Module / Home

API 0 devnetuser

DASHBOARD SYSTEM INFO

DEVICE INVENTORY

13 Devices

- Managed (1)
- In-Progress (11)
- Collection Failure (1)

DISCOVERY - UNREACHABLE DEVICES

4 Devices

BRANCH SITES

To set up branch sites, use IWAN app!

NETWORK PLUG AND PLAY PROJECTS

4

- 1 Provisioned
- 3 Pre-Provisioned
- 0 In-Progress

EASYQOS SCOPES

All Path traces are successful.

I wish this page would...

- **Servicios:** APIC-EM descubre los dispositivos en la red y los coloca en tablas de inventario de host y dispositivos donde se pueden ver los detalles de cada dispositivo. El APIC-EM también crea una topología.
- **Aplicaciones:** APIC-EM permiten la configuración de la red y la prueba de conectividad de red. Las aplicaciones incluyen:
 - **IWAN:** simplifica las implementaciones de WAN al proporcionar una interfaz intuitiva basada en políticas que ayuda a TI a abstraer la complejidad de la red y el diseño para la intención comercial.
 - **Path Trace:** facilita y acelera enormemente la tarea de supervisión de la conexión y solución de problemas.
 - **Network Plug and Play:** proporciona un enfoque unificado para aprovisionar redes empresariales compuestas de enrutadores, conmutadores y puntos de acceso inalámbrico de Cisco con una experiencia de implementación casi nula.
 - **Easy QoS:** proporciona una forma sencilla de clasificar y asignar prioridad de aplicación.
- **Documentación de API:** es un enlace donde puede acceder a información sobre la API que es esencial para los programadores de red.

Esta es la topología descubierta y renderizada por APIC-EM probablemente puede variar o coincidir con la que se muestra en la imagen.

The screenshot displays the Cisco APIC-EM interface for the 'Topology' section. The left sidebar contains navigation options: Home, Discovery, Device Inventory, Host Inventory, Topology (selected), IWAN, Path Trace, Network Plug and Play, and EasyQoS. The main area shows a network topology diagram with the following components and connections:

- cloud node** (cloud icon) at the top, connected to two **2 Routers** (purple icons).
- The two **2 Routers** are connected to a **Campus-WLC-5506** (green icon).
- The **Campus-WLC-5506** is connected to **2 Switches and Hubs** (yellow icons).
- One of the **2 Switches and Hubs** is connected to **CAMPUS-Dist2 Branch-Access1** (yellow icon).
- The **CAMPUS-Dist2 Branch-Access1** is connected to another **2 Switches and Hubs** (yellow icon).
- This second **2 Switches and Hubs** is connected to an **AP7061-DSW-190a** (orange icon).
- The **AP7061-DSW-190a** is connected to a wireless device icon labeled **10.1.15.117**.
- Another **2 Switches and Hubs** (yellow icon) is connected to two desktop computer icons labeled **10.1.12.20** and **10.1.1.22**.
- A separate **2 Routers** (purple icon) is shown on the right, labeled **Branch2-Router.yourdomain.com**.

The interface includes a search bar 'Search Topology...', a navigation menu with icons for zooming and editing, and a user profile 'devnetuser' with a notification bell showing '0'.

I wish this page would...

Swagger es un marco para el desarrollo y la documentación de API RESTful bajo la especificación OpenAPI. Proporciona una forma estándar de documentar API que es consistente en presentación y funcionalidad.

El APIC-EM utiliza un token para la autenticación. Los tokens se llaman tickets de servicio. Para solicitar un ticket de servicio, envía una solicitud POST al punto final API / ticket.

1. Haga clic en **API** en la esquina superior derecha de la pagina de inicio.
2. Haga clic en **Role Based Access Control>ticket>POST/ticket**. Debería ver la documentación del método para crear un nuevo ticket de usuario, como se muestra en la imagen.



Available APIs

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



Role Based Access Control

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

[Terms of service](#)

[Cisco DevNet](#)

aaa : APIs to register and manage AAA Servers

Show/Hide | List Operations | Expand Operations | Raw

role : Role Description API

Show/Hide | List Operations | Expand Operations | Raw

ticket : Ticket Management API



Show/Hide | List Operations | Expand Operations | Raw

user : User Management API

Show/Hide | List Operations | Expand Operations | Raw

[BASE URL: <https://sandboxapicem.cisco.com/api/v1/api-docs/rbac-service> , API VERSION: 1.0]

3. Haga clic en **Model Shema**.
4. Seleccione el cuadro Amarillo debajo de Model Shema.
5. Ingrese las credenciales como, por ejemplo:
 - **username:** devnetuser
 - **password:** Cisco123!
6. Haga clic en **Try it out!**
7. Si tiene éxito, el número del ticket estará en el cuerpo de respuesta JSON, como se muestra en la siguiente imagen.

POST /ticket

addTicket

Implementation Notes

This method is used to create a new user ticket

Response Class

Model | Model Schema

TicketRbacResult {

version (*string, optional*),
 response (*ServiceTicketRbac, optional*)

}

ServiceTicketRbac {

idleTimeout (*integer, optional*),
 sessionTimeout (*integer, optional*),
 serviceTicket (*string*): Service Ticket to be used as authentication Ticket

}

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
user	<pre>{ "password": "Cisco123!", "username": "devnetuser" }</pre>	user	body	Model Model Schema

Parameter content type:

Click to set as parameter value

Error Status Codes

HTTP Status Code	Reason
200	This Request is OK
202	This Request is Accepted
403	This user is Forbidden Access to this Resource
401	Not Authorized Yet, Credentials to be supplied
404	No Resource Found

[Try it out!](#)[Hide Response](#)

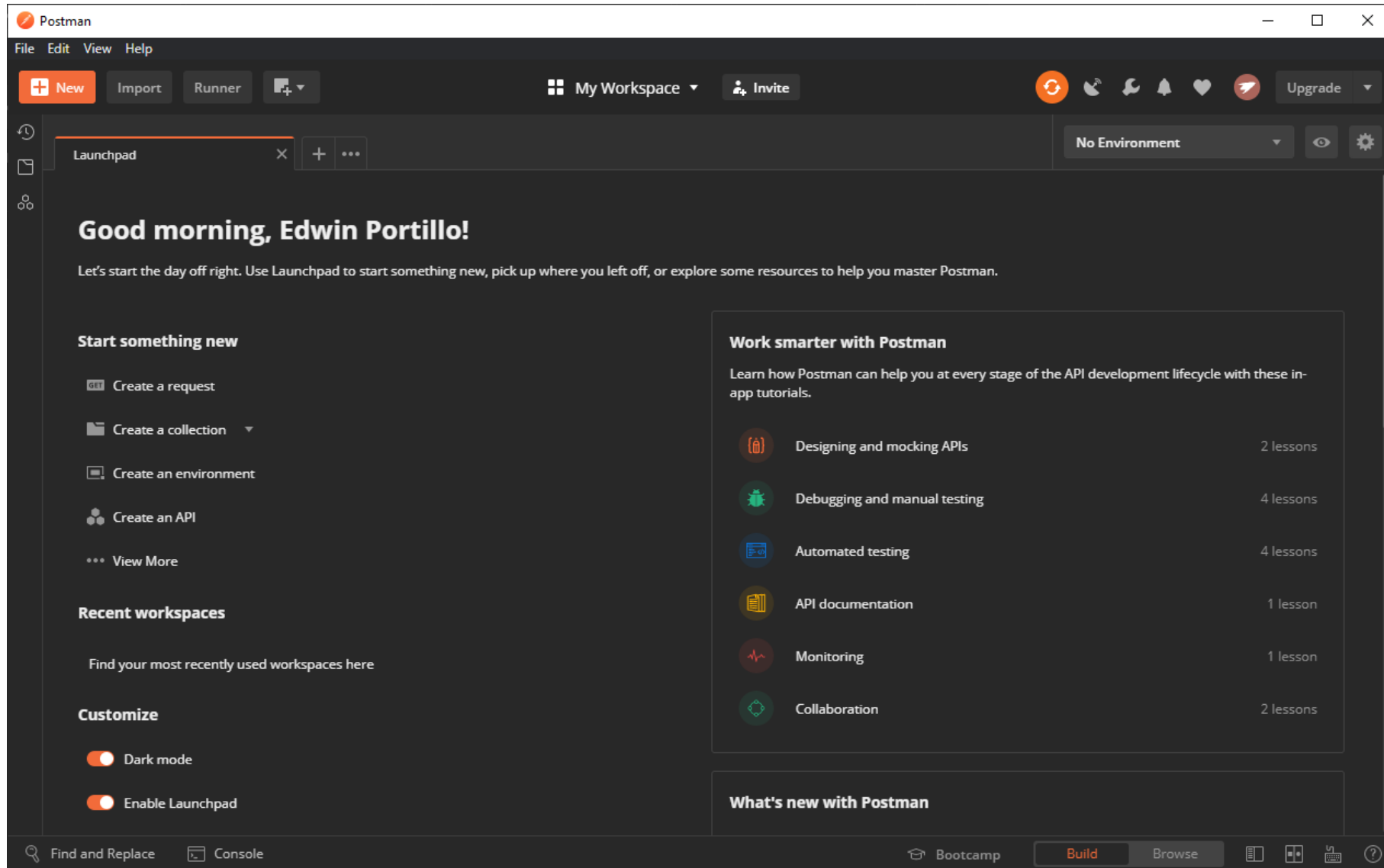
Request URL

```
https://sandboxapicem.cisco.com/api/v1/ticket
```

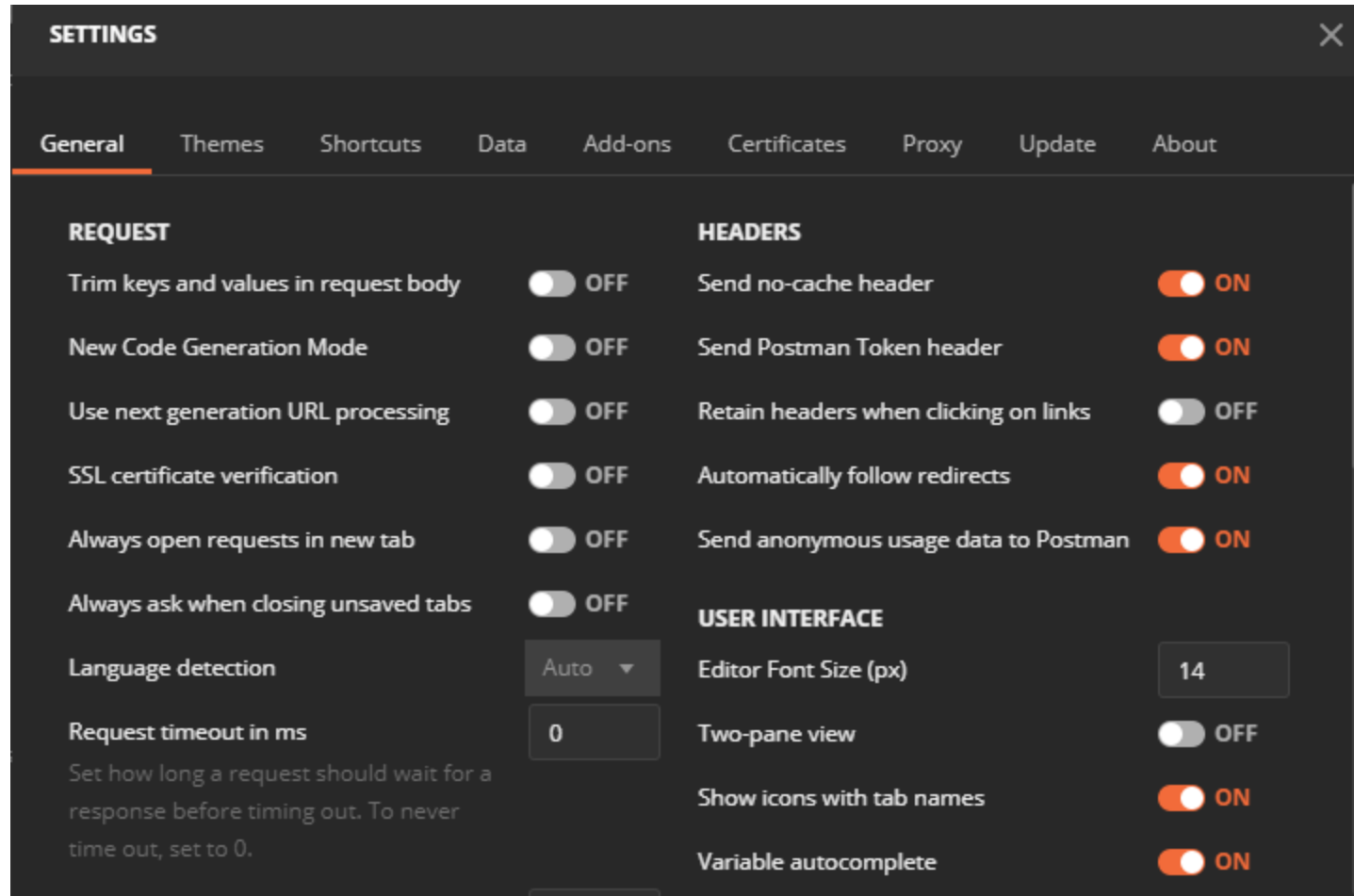
Response Body

```
{"response": {"serviceTicket": "8T-18585-eDgcJ8glruKFpFc7fXiE-cas", "idleTimeout": 1800, "sessionTimeout": 21600, "version": "1.0"}}
```

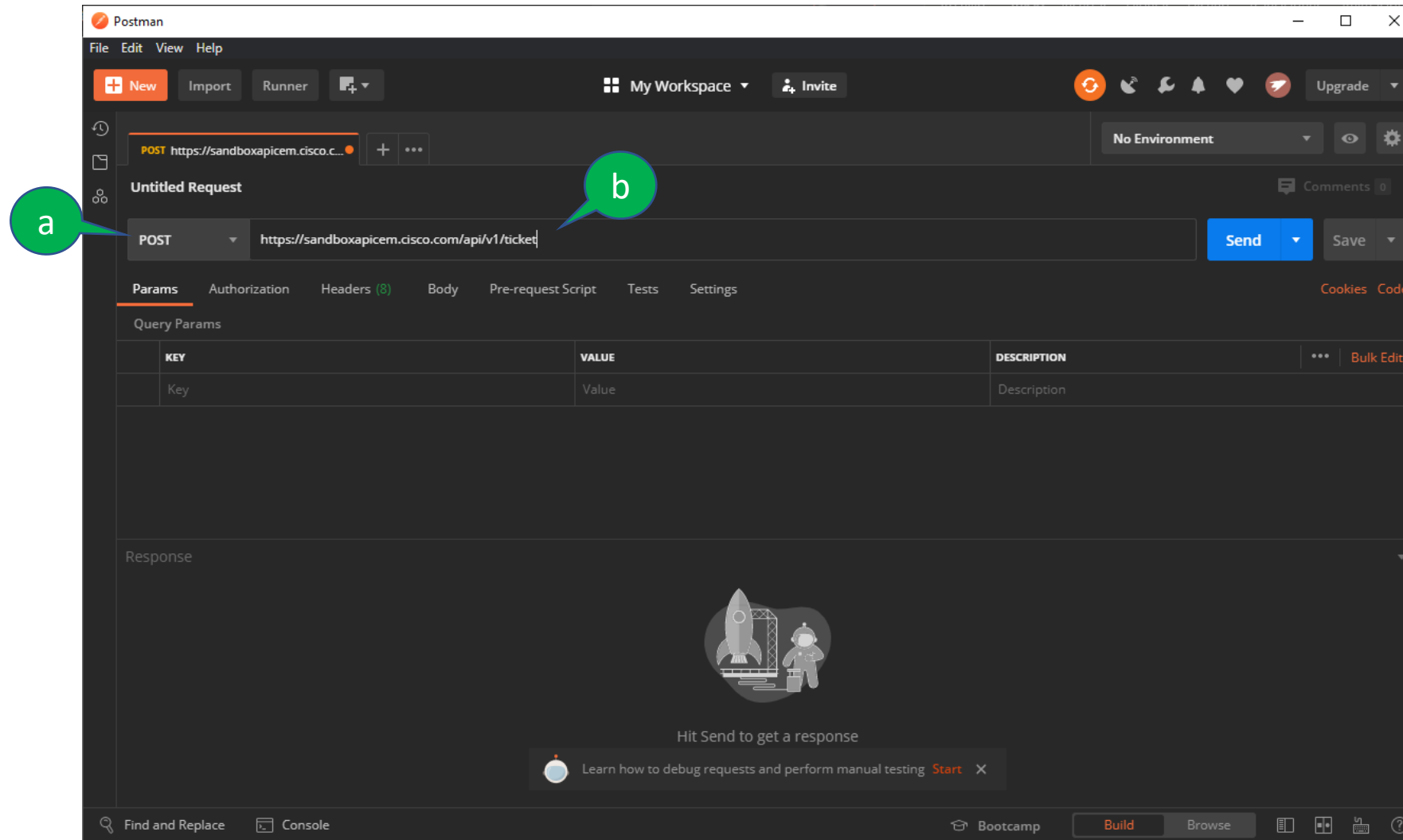
Descargue e instale Postman <https://www.postman.com/downloads/>



1. Inicie sesión en **Postman**.
2. Haga clic en **File>Settings>General** y deshabilite **SSL certificate verification**.



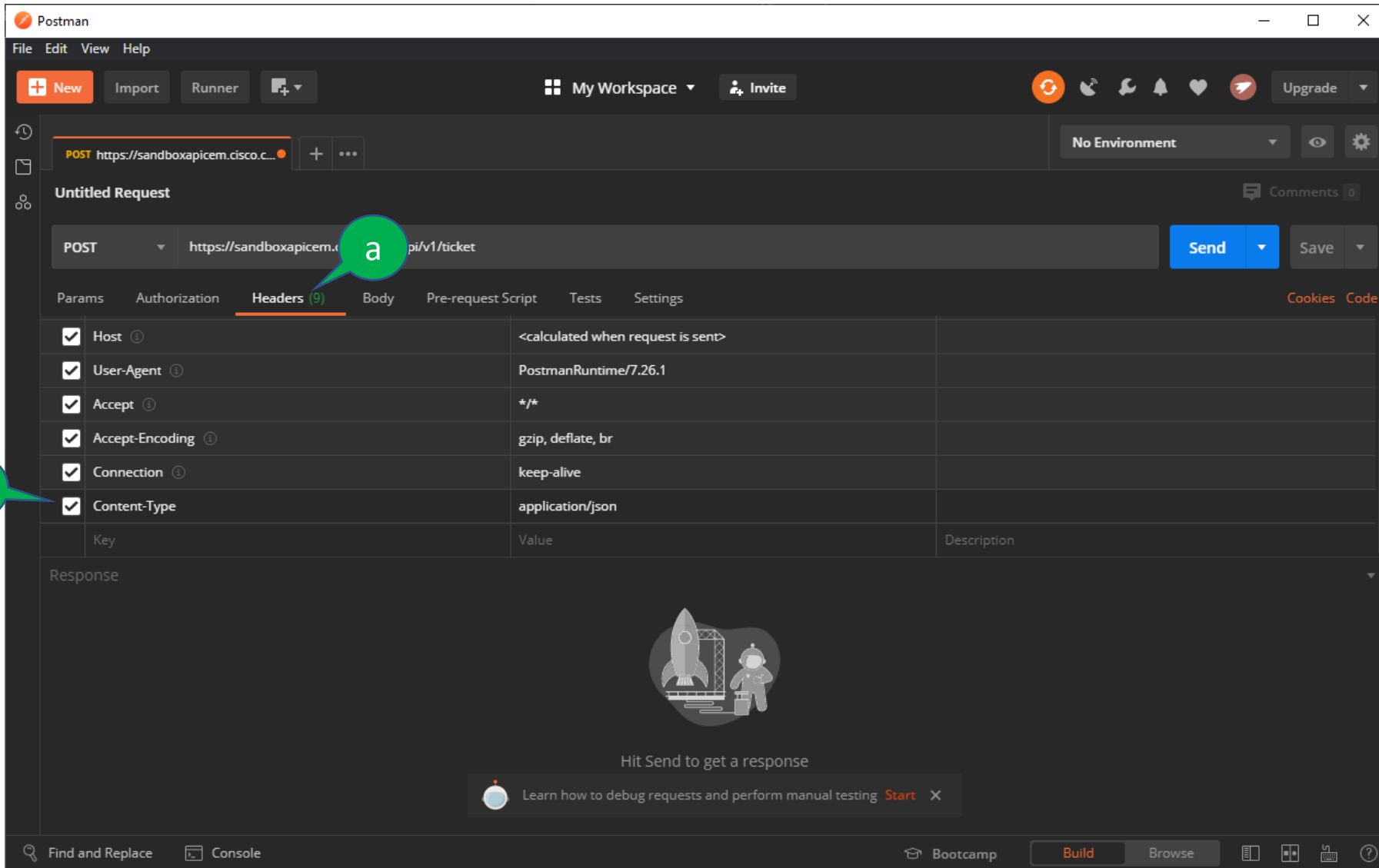
3. Selección del método e ingreso de la URL requerida.



Método: **POST**

URL: **https://sandboxapicem.cisco.com/api/v1/ticket**

3. Ingrese la información del encabezado.



Si no esta creada la llave debe crearla. Content-Type: **application/json**

4. Ingrese la información de la autenticación **Body** > **raw**.

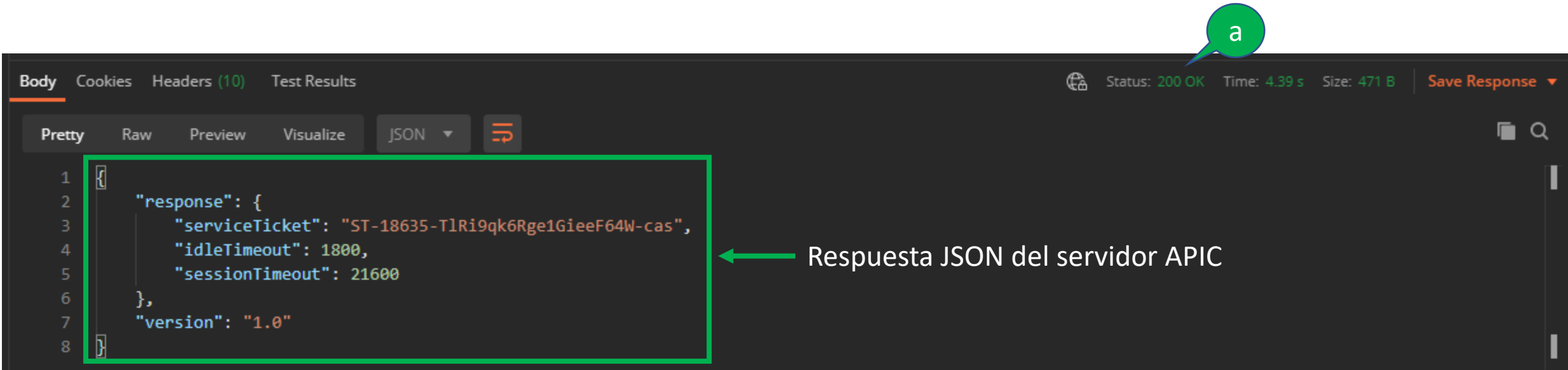
The screenshot shows the Postman application window. At the top, there's a menu bar (File, Edit, View, Help) and a toolbar with buttons for 'New', 'Import', 'Runner', and 'Upgrade'. Below that, the workspace shows 'My Workspace' and 'No Environment'. The main area is titled 'Untitled Request' and shows a POST request to 'https://sandboxapicem.cisco.com/api/v1/'. The 'Body' tab is selected, and the 'raw' radio button is chosen. The request body is a JSON object:

```
1 {
2   "username": "devnetuser",
3   "password": "Cisco123!"
4 }
5
```

 A green arrow points from the JSON in the right margin to the 'raw' body field in the interface. The 'Response' section at the bottom shows a message: 'Hit Send to get a response'.

{
"username": "devnetuser",
"password": "Cisco123!"
}

5. Haga clic en el botón **Send**. Después de unos segundos deberá ver la respuesta JSON con el valor de la clave **serviceTicket**.



The screenshot shows a REST client interface with the following details:

- Top bar: Status: 200 OK, Time: 4.39 s, Size: 471 B, Save Response
- Navigation tabs: Body (selected), Cookies, Headers (10), Test Results
- View options: Pretty (selected), Raw, Preview, Visualize, JSON
- JSON response body (lines 1-8):

```
1 {
2   "response": {
3     "serviceTicket": "ST-18635-TlRi9qk6Rge1GieeF64W-cas",
4     "idleTimeout": 1800,
5     "sessionTimeout": 21600
6   },
7   "version": "1.0"
8 }
```
- A green box highlights the response body.
- A green arrow points to the box with the text "Respuesta JSON del servidor APIC".
- A green circle with the letter "a" is in the top right corner.

El valor devuelto del **serviceTicket** puede ser diferente al que se muestra en la imagen no necesariamente tienen que coincidir lo importante es que el estado de respuesta sea 200 OK.

Listar el inventario *de los hosts*



Postman se configurará para que envíe solicitudes a APIC y así generar el inventario de los hosts.

1. Haga clic en el signo más (+) para crear una nueva pestaña en Postman.
2. Ingrese la siguiente información:
 - Método: **GET**
 - URL: **https://sandboxapicem.cisco.com/api/v1/host**
 - Encabezados
 - **Content-Type:application/json**
 - **X-Auth-Token:** <déjelo en blanco>
3. Haga clic en la pestaña de la solicitud GET de servicio que creó anteriormente y copie el valor del ticket del servicio, sin comillas de la respuesta JSON.
4. Regrese a la pestaña de la solicitud del inventario de host y pegue el valor del ticket en el campo del valor **X-Auth-Token**.
5. Haga clic en **Send** y verifique que en la sección del Body se complete con datos JSON como se muestra en la siguiente imagen.

POST https://sandboxapicem.cisco.c... GET https://sandboxapicem.cisco.co...

No Environment

Untitled

GET https://sandboxapicem.cisco.com/api/v1/host/

Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Key	Value
<input checked="" type="checkbox"/> Content-Type	application/json
<input checked="" type="checkbox"/> X-Auth-Token	ST-18686-kbnwalkosge6y7gyOZ6f-cas

Body Cookies Headers (11) Test Results

Status: 200 OK Time: 293 ms Size: 1.8 KB Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "response": [
3     {
4       "hostIp": "10.1.15.117",
5       "hostMac": "00:24:d7:43:59:d8",
6       "hostType": "wireless",
7       "connectedNetworkDeviceId": "cd6d9b24-839b-4d58-adfe-3fdf781e1782",
8       "connectedNetworkDeviceIpAddress": "10.1.14.3",
9       "connectedAPMacAddress": "68:bc:0c:63:4a:b0",
10      "connectedAPName": "AP7081.059f.19ca",
11      "vlanId": "600",
12      "lastUpdated": "1479514114932",
13      "source": "200",
14      "pointOfPresence": "ae19cd21-1b26-4f58-8ccd-d265deabb6c3",
15      "pointOfAttachment": "ae19cd21-1b26-4f58-8ccd-d265deabb6c3",
16      "subType": "UNKNOWN",
17      "id": "48cdeb9b-b412-491e-a80c-7ec5bbe98167"
18    },
19    {
20      "hostIp": "10.2.1.22",
21      "hostMac": "5c:f9:dd:52:07:78",
22      "hostType": "wired",
23      "connectedNetworkDeviceId": "26450a30-57d8-4b56-b8f1-6fc535d67645",
24      "connectedNetworkDeviceIpAddress": "10.2.1.17",
25      "connectedInterfaceId": "7075521d-4d7b-4218-92f9-79b60e054635",
26      "connectedInterfaceName": "GigabitEthernet1/0/47",
27      "vlanId": "200",
28      "lastUpdated": "1479514299803",
29      "source": "200",
30      "subType": "UNKNOWN",

```

f

a

d

b

c

e

Utilice CodeBeautify para evaluar la respuesta.

6. Copie el resultado de la solicitud hecha en Postman a JSON Viewer

<https://codebeautify.org/jsonviewer>

7. Haga clic en **Tree Viewer** para que muestre la información en formato de árbol, así:

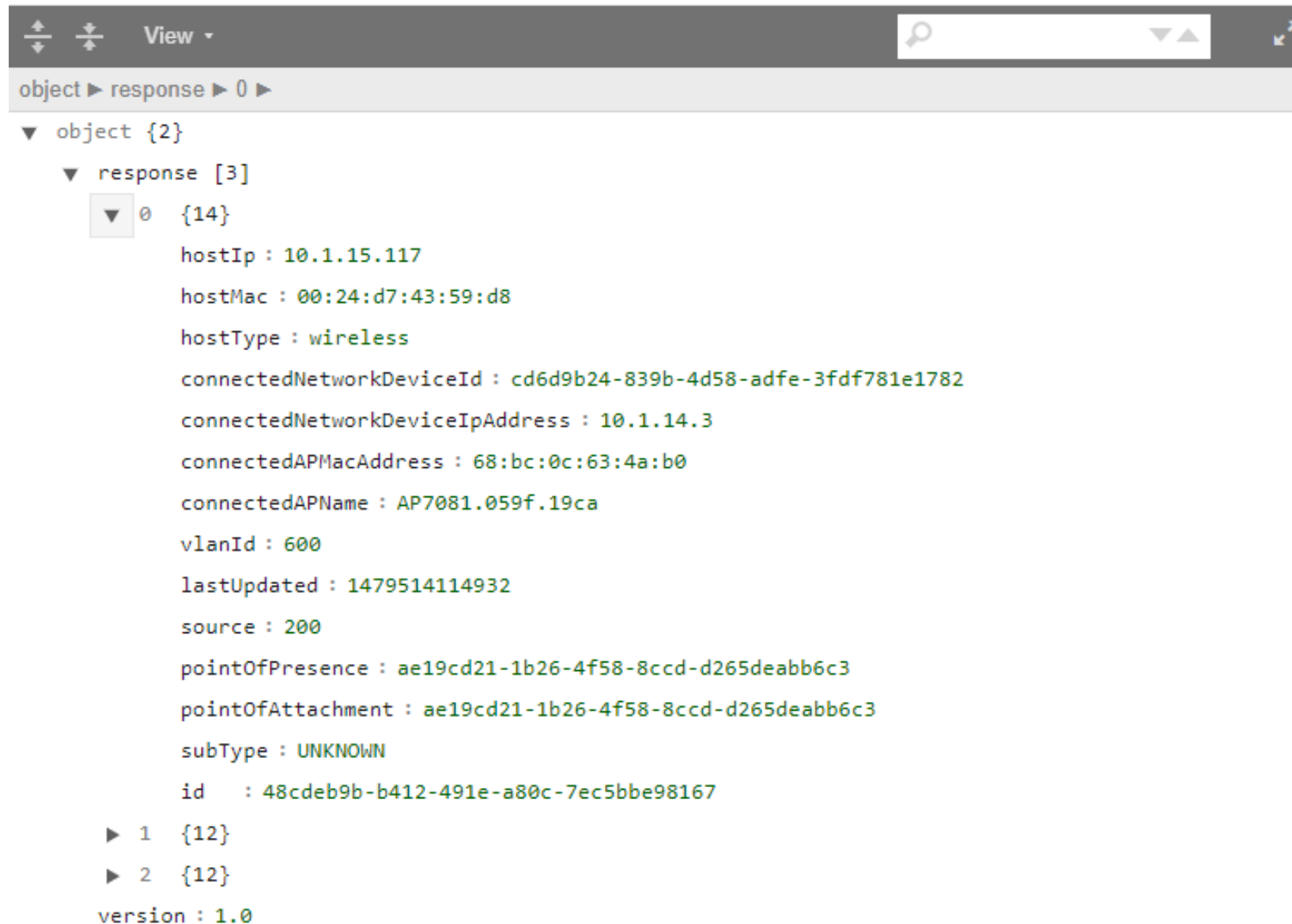
The screenshot displays the Code Beautify JSON Viewer interface. The left pane shows the raw JSON data with line numbers 6 through 49. The middle pane contains a control panel with the following elements:

- Result mode:** A dropdown menu set to "tree".
- Buttons:** "Load Url", "Browse", "Tree Viewer" (highlighted with a green circle 'b'), "Beautify", "Minify", "Validate", "JSON to XML", "JSON to CSV", and "Download".
- Advertisement:** A box for "Generador Formularios PHP" with an "OPEN" button.

The right pane shows the JSON data in a tree view format, with a green circle 'c' highlighting the "View" dropdown menu. The tree structure is as follows:

```
object ▶ response ▶ 1 ▶  
  ▼ object {2}  
    ▼ response [3]  
      ▼ 0 {14}  
        hostIp : 10.1.15.117  
        hostMac : 00:24:d7:43:59:d8  
        hostType : wireless  
        connectedNetworkDeviceId : cd6d9b24-839b-4d58-adfe-3fdf781e1782  
        connectedNetworkDeviceIpAddress : 10.1.14.3  
        connectedAPMacAddress : 68:bc:0c:63:4a:b0  
        connectedAPName : AP7081.059f.19ca  
        vlanId : 600  
        lastUpdated : 1479514114932  
        source : 200  
        pointOfPresence : ae19cd21-1b26-4f58-8ccd-d265deabb6c3  
        pointOfAttachment : ae19cd21-1b26-4f58-8ccd-d265deabb6c3  
        subType : UNKNOWN  
        id : 48cdeb9b-b412-491e-a80c-7ec5bbe98167  
      ▼ 1 {12}  
        hostIp : 10.2.1.22  
        hostMac : 5c:f9:dd:52:07:78  
        hostType : wired  
        connectedNetworkDeviceId : 26450a30-57d8-4b56-bf1-6fc535d67645  
        connectedNetworkDeviceIpAddress : 10.2.1.17  
        connectedInterfaceId : 7075521d-4d7b-4218-92f9-79b60e054635  
        connectedInterfaceName : GigabitEthernet1/0/47  
        vlanId : 200  
        lastUpdated : 1479514299803  
        source : 200  
        subType : UNKNOWN  
        id : f624d4f3-0ab9-4ae3-b09d-62051edbd8f3
```

Observe los pares **clave:valor** asignados al nivel 0. Junto al nivel 0, el número 14 indica que hay 14 claves asociadas con esta entrada.



```
object > response > 0 >
  ▼ object {2}
    ▼ response [3]
      ▼ 0 {14}
        hostIp : 10.1.15.117
        hostMac : 00:24:d7:43:59:d8
        hostType : wireless
        connectedNetworkDeviceId : cd6d9b24-839b-4d58-adfe-3fdf781e1782
        connectedNetworkDeviceIpAddress : 10.1.14.3
        connectedAPMacAddress : 68:bc:0c:63:4a:b0
        connectedAPName : AP7081.059f.19ca
        vlanId : 600
        lastUpdated : 1479514114932
        source : 200
        pointOfPresence : ae19cd21-1b26-4f58-8ccd-d265deabb6c3
        pointOfAttachment : ae19cd21-1b26-4f58-8ccd-d265deabb6c3
        subType : UNKNOWN
        id : 48cdeb9b-b412-491e-a80c-7ec5bbe98167
      ▶ 1 {12}
      ▶ 2 {12}
    version : 1.0
```

Podría utilizar un lenguaje de programación como Python para hacer las llamadas a la API con JSON para generar el inventario de dispositivos o una copia de seguridad de las configuraciones.

Bibliografía

Cisco. (Agosto de 2020). Controlador de infraestructura de políticas de aplicaciones (APIC) de Cisco. Obtenido de https://www.cisco.com/c/es_mx/products/cloud-systems-management/application-policy-infrastructure-controller-apic/index.html

DevNet. (Agosto de 2020). Sandbox Labs. Obtenido de <https://developer.cisco.com/site/sandbox/>