2.3.2.1: Lab - Create a Host Inventory in Python-2

This page summarizes the steps in **Lab - Create a Host Inventory in Python**. Click **here (https://288647296.netacad.com/courses/951643/files/82148700/download?wrap=1)**  **(https://288647296.netacad.com/courses/951643/files/82148700/download?wrap=1)** to download the lab PDF if you would like a more detailed explanation.

In this lab, you will complete the following objectives in order to prepare your computer for the course:

- **Part 1: Use Postman to get a Network Host Inventory**
- **Part 2: Use Python to get a Network Host Inventory**

Required Resources

- Postman
- Python 3 with IDLE
- Python **requests** module
- Python **tabulate** module
- The functions file that you have created or the **apic_em_functions_sol.py** file
- Access to the Internet

Note: Use the APIC-EM sandbox URL and credentials provided by your instructor. Only use the public APIC-EM URL and credentials for additional study after the conclusion of the workshop. For example purposes, this activity uses the URL and credentials of the public sandbox.

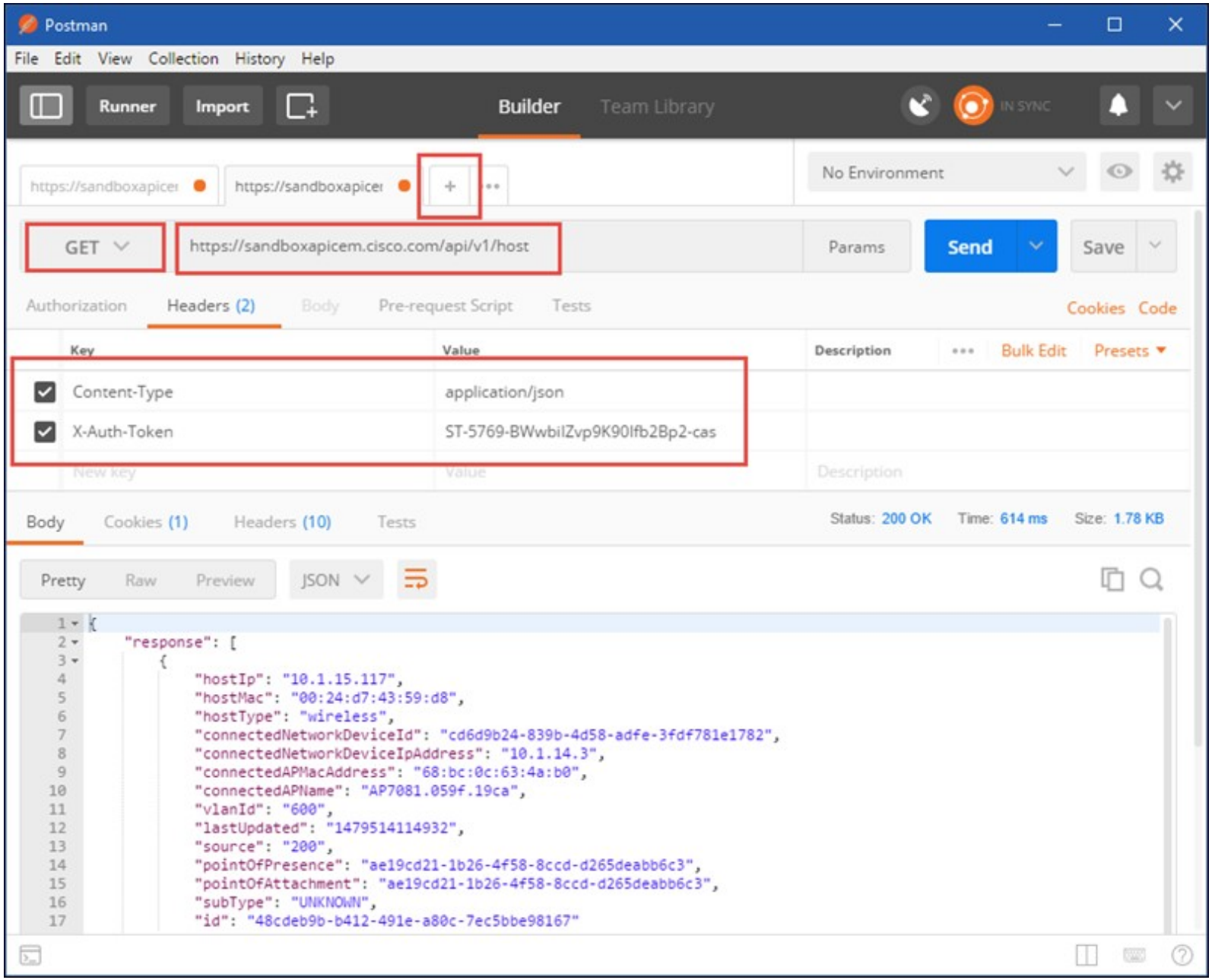
Part 1: Use Postman to get a Network Host Inventory

In this part of the lab, you will enter the requirements for accessing the API in Postman, submit a request, and then review the JSON data that is returned.

Step 1: Configure and send the Postman request for a host inventory.

Refer to the Postman screen shot below and complete the following steps:

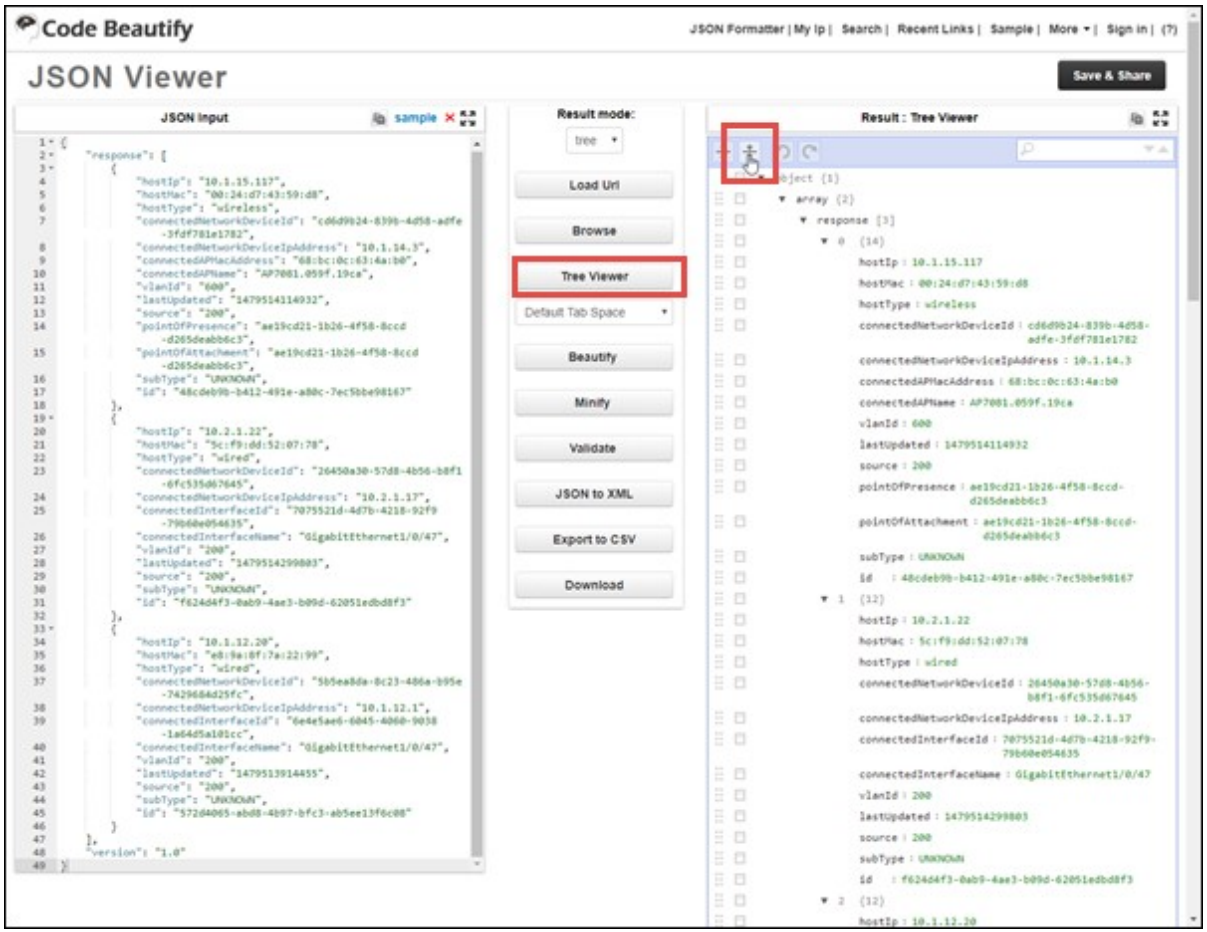
a. Click the plus sign (+) to create a new tab.
b. Enter the following information:
   - Request method: **GET**
   - Endpoint URI: **https://{YOUR-APICEM}.cisco.com/api/v1/host**
   - Headers:
     Content-Type: application/json
     X-Auth-Token: <leave this blank for now>
c. Click the tab for the service ticket request you created in the previous lab and click **Send**.
d. Copy the value of the service ticket, without quotes, from the response JSON.
e. Return to the host inventory request tab and paste the value of the service ticket into the **Value** field for the **X-Auth-Token**.
f. Click **Send** and verify the body section is populated with JSON data, as shown below. If the response fails, look at the status value and try to determine where the error may be.



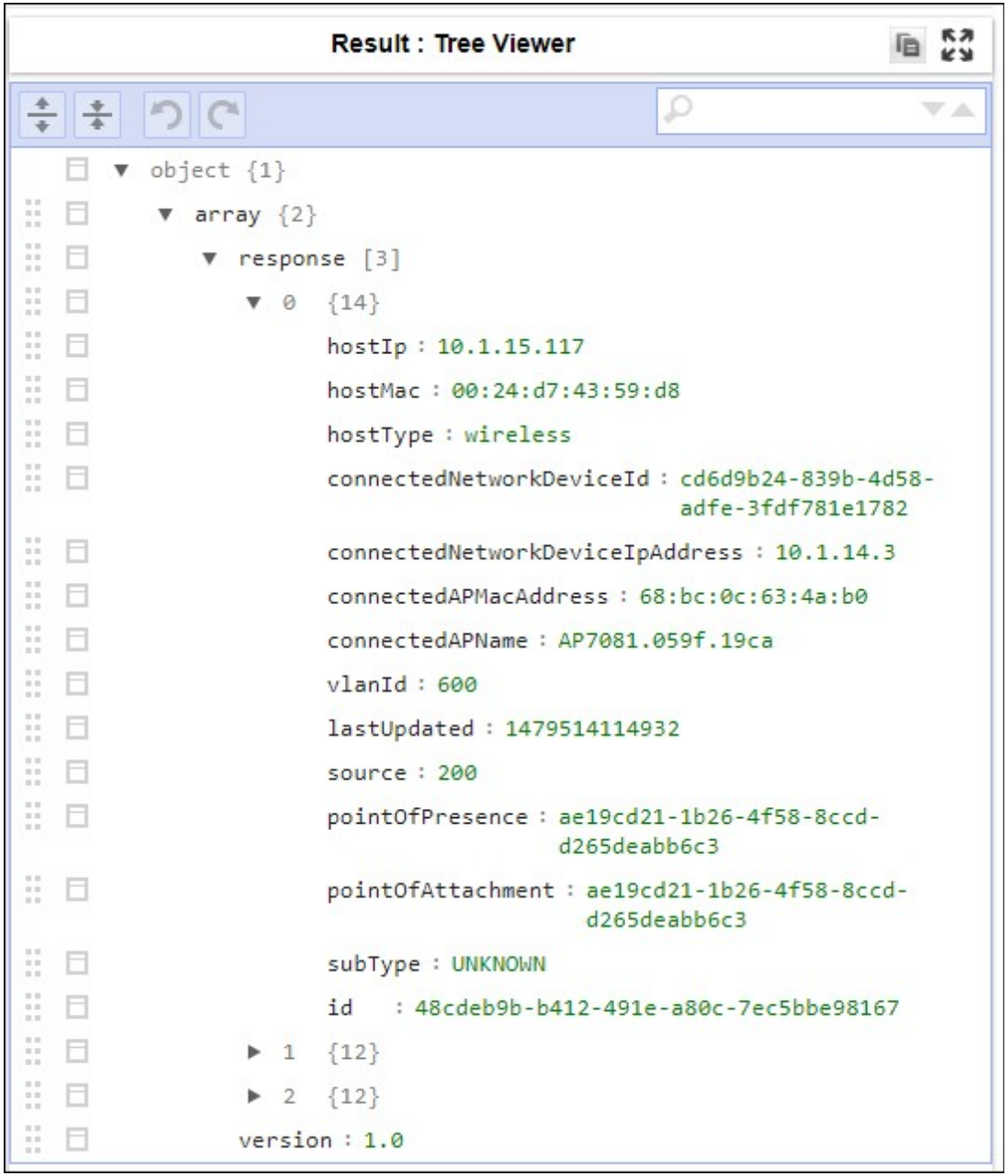
Step 2: Use CodeBeautfiy.com to evaluate the response.

a. Copy the JSON from Postman to JSON Viewer at **https://codebeautify.org/jsonviewer (https://codebeautify.org/jsonviewer)** .

b. Click **Tree Viewer** to render the tree, as shown below:



c. Collapse all levels by clicking the second icon in the Result window, as shown above. Expand **object**, **array**, and **response**. The number next to the response key indicates how many entries there are. Expand the level marked **0**. Your Tree Viewer should look similar to the following:



d. Look at the key/value pairs that are assigned to level 0. Next to level 0, the number 14 indicates that there are 14 keys associated with this entry. However, there are different numbers for the other hosts. Open each

device and compare entries.

Part 2: Use Python to get a Network Host Inventory

In this part of the lab, you will create a Python program to get the same network host inventory you retrieved using Postman in Part 1. You will then create a function from your program and add it to your **my_apic_em_functions.py** file.

Step 1: Setup the code environment.

1. Open IDLE and click **File > New File**. Save the new file as **print_hosts.py**.
2. Enter the following code to import the **requests**, **json**, **tabulate**, and the functions from **my_apic_em_functions.**

```
import requests
import json
from tabulate import *
from my_apic_em_functions import *
```

Step 2: Build the request components.

a. Create the variable **api_url** and assign a string containing the URI of the APIC-EM **/host** endpoint. The URL is **https://{YOUR-APICEM}.cisco.com/api/v1/host**.
b. Create the variable **ticket** and assign to it the value returned by the **get_ticket()** function.
c. Create the **headers** dictionary and assign it to the **headers** variable.

```
api_url = "https://sandboxapicem.cisco.com/api/v1/host"
ticket = get_ticket()
headers = {
 "content-type": "application/json",
 "X-Auth-Token": ticket
}
```

Step 3: Make the request and handle errors.

a. Create a variable named **resp**, and assign to it the results of request. You use the **get()** method to make the request by supplying it with the URL and headers variables created above.
b. Print the status of the request.
c. Create an **if** condition to detect if the status code returned by the API is anything other than 200. If this condition was true, the request was unsuccessful. If the request fails, raise an exception and provide the error message as shown in the code block below.
d. Create a variable to hold the response JSON that has been converted to Python dictionary format.

The entire block of code should look like this:

```
resp = requests.get(api_url, headers=headers, verify=False)
print("Status of /host request: ", resp.status_code)
if resp.status_code != 200:
    raise Exception("Status code does not equal 200. Response text: " + resp.text)
response_json = resp.json()
```

e. Save your script and run it. You should get output similar to the following. If not, troubleshoot your code for errors.

```
The service ticket number is: ST-5873-WBA3XaHjev0aJYeWPVk3-cas
Status of /host request: 200
>>>
```

Step 4: Parse and format the JSON response data.

In this step, you will create a **for** loop to parse the JSON data and create a table that will look similar to the following:

```
  Number  Type      IP
--------  --------  -----------
       1  wireless  10.1.15.117
       2  wired     10.2.1.22
       3  wired     10.1.12.20
```

To do this, a **for** loop will iterate through the list of hosts and extract the value for the two dictionary keys, **hostType** and **hostIP**. The **for** loop will populate a list called **host_list** with the values for the Number, Type, and IP columns. The number of the host is not present in the JSON data. You will create a separate variable, assign a value of 0 and then increment it as the loop repeats.

a. Create the list variable **host_list** to hold the **hostType** and **hostIP** values and create the integer variable **i** that will be incremented with each iteration of the loop. The **i** variable will hold the value for the number column in the table.
b. Create the **for** loop to iterate over every item in the response key of the **response_json** variable.
c. Inside the **for** loop, increment the ordinal number variable **i.** Then append the values from the JSON to the **host_list** variable.
d. Use the **tabulate** function to print the table of hosts. The tabulate function will take as arguments the **host_list** variable, and a list variable of headers for the columns that will be printed.
e. Save your script and run it. You should get a table similar to the one shown at the beginning of this step. Investigate and fix any errors that may occur. The correct code is as follows:

```
host_list = []
i = 0
for item in response_json["response"]:
     i+=1
     host = [
              i,
              item["hostType"],
              item["hostIp"]
             ]
     host_list.append( host )
table_header = ["Number", "Type", "IP"]
print( tabulate(host_list, table_header) )
```

Step 5: Create the function for the host inventory request.

a. The **my_apic_em_functions.py** should already have the import statements for **requests** and **json**. Copy and paste the **tabulate** import statement into the file below them.

b. Add a few blank lines below your **get_ticket():** function and define a new function called **print_hosts()**: using the **def** command.
c. Copy your code from the **api_url** variable through the **print** statement and paste it into the new function.
d. Select all of the lines below the definition statement and select **Indent Region** from the IDLE **Format** menu.
e. Your code should look like the following:

my_apic_em_functions.py - C:\APIC-EM Labs\my_apic_em_functions.py (3.6.4)

File Edit Format Run Options Window Help

```
import requests
import json
from tabulate import *

requests.packages.urllib3.disable_warnings()  # Disable SSL warnings

def get_ticket():
    api_url = "https://sandboxapicem.cisco.com/api/v1/ticket"
    headers = {
        "content-type": "application/json"
    }
    body_json = {
        "username": "devnetuser",
        "password": "Cisco123!"
    }
    resp = requests.post(api_url, json.dumps(body_json), headers=headers, verify=False)
    status = resp.status_code  # status code property of resp object
    response_json = resp.json()
    serviceTicket = response_json["response"]["serviceTicket"]
    print("The service ticket number is: ", serviceTicket)
    return serviceTicket

def print_hosts():
    api_url = "https://sandboxapicem.cisco.com/api/v1/host"
    ticket = get_ticket()
    headers = {
        "content-type": "application/json",
        "X-Auth-Token": ticket
    }
    resp = requests.get(api_url, headers=headers, verify=False)
    response_json = resp.json()
    host_list = []
    i = 0
    for item in response_json["response"]:
        i += 1
        host = [
                i,
                item["hostType"],
                item["hostIp"]
               ]
        host_list.append( host )
    table_header = [
                    "Number",
                    "Type",
                    "IP"
                   ]
```