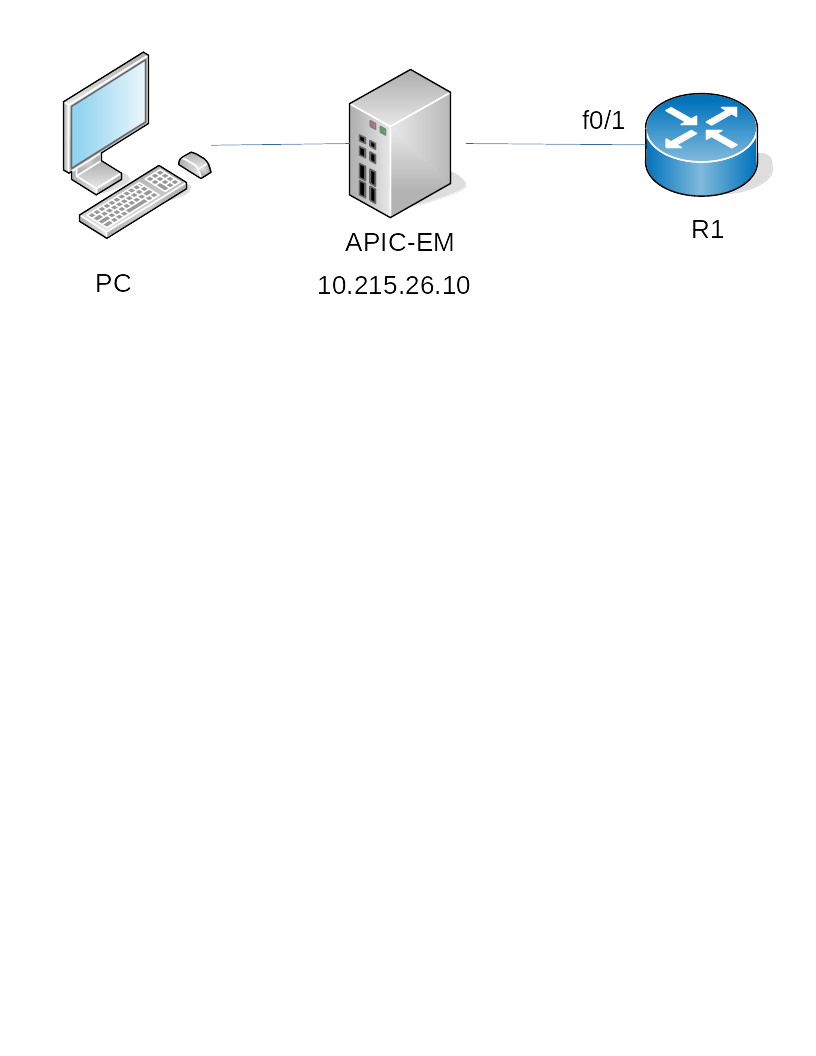
LAB 7 : Dùng chương trình Postman và chương trình python lấy ticket, list of devices trên APIC-EM

1. **Sơ đồ mạng:**



Hình1.1: Sơ đồ bài Lab

1. **Mô tả :**

* Sơ đồ gồm 1 APIC-EM server , 1 PC và 1 Cisco Router được kết nối với nhau như hình 1.1.
* Trên sơ đồ này, học viên thực hiện lấy ticket và danh sách thiết bị bằng phần mềm Postman và bằng ngôn ngữ Python
* Máy PC phải đáp ứng yêu cầu đã cài đặt trạm làm việc cho developer

1. **Yêu cầu kỹ thuật:**

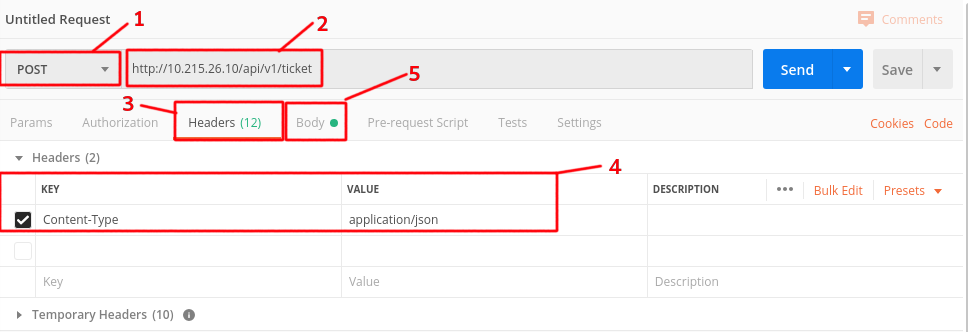
* Học viên thực hiện đấu nối thiết bị, thực hiện một số cấu hình cơ bản trên Router như đặt hostname, password console, đặt địa chỉ IP động.
* Thực hiện cấu hình ping thành công đến APIC-EM server
* Dùng Postman lấy ticket và danh sách thiết bị trên APIC-EM
* Viết chương trình bằng Python thực hiện lấy ticket và danh sách thiết bị trên APIC-EM

1. **Các bước thực hiện :**

**POSTMAN :**

**Phần 1: Get Ticket**

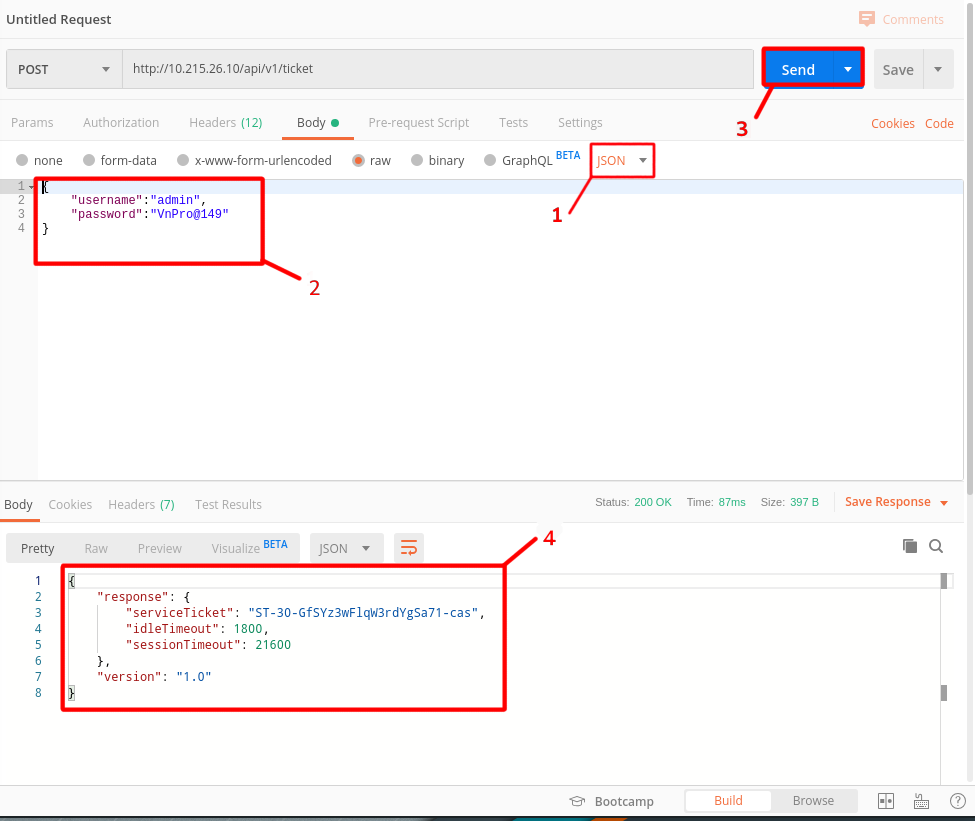
Khởi động Postman



* Chọn chế độ POST ở **[1]**
* Nhập địa chỉ APIC-EM ở **[2]** : Cú pháp của URL

*http://[địa chỉ IP APIC-EM]/api/v1/[REST function ]*

(Trong trường hợp này để lấy ticket nên REST function sẽ là /ticket)

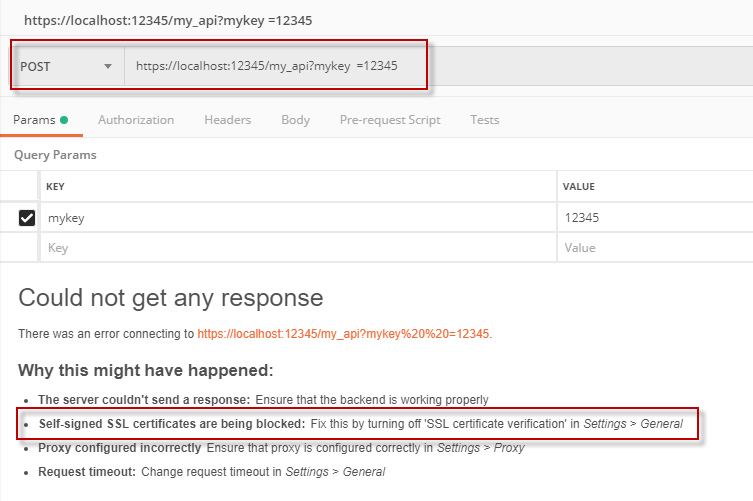
* Chuyển sang tab Headers **[3]** để thêm HTTP header
* Hàm POST bắt buộc phải được định danh key “Content:Type” với giá trị là “Application/json” như ở mục **[4]**
* Khi đã làm hết các bước trên ta chuyển sang tab Body **[5]** để khai báo Username/password ở định dạng Json
* Chọn định dạng Json ở **[1]**
* Khai báo Username/Password ở **[2]** theo định dạng Json có dạng như trên

Một số lưu ý hàm Json:

* 1. Giữa giá trị Key và value được ngăn cách bởi “**:**”
  2. Key phải đặt trong “ “
  3. Giữa các Key được ngăn cách bởi dấu “**,**”
  4. Các key phải nằm trong { }
* Khi khai báo xong Username/Password thì nhấn Send **[3],** Nếu status trả về là “200 OK” thì thành công, ticket sẽ trả về ở mục **[4]**

Ticket sẽ là “ST-30-GfSYz3wFlqW3rdYgSa71-cas”

Nếu bị lỗi sau vào file → Setting → General tắt mục SSL certificate verification đi

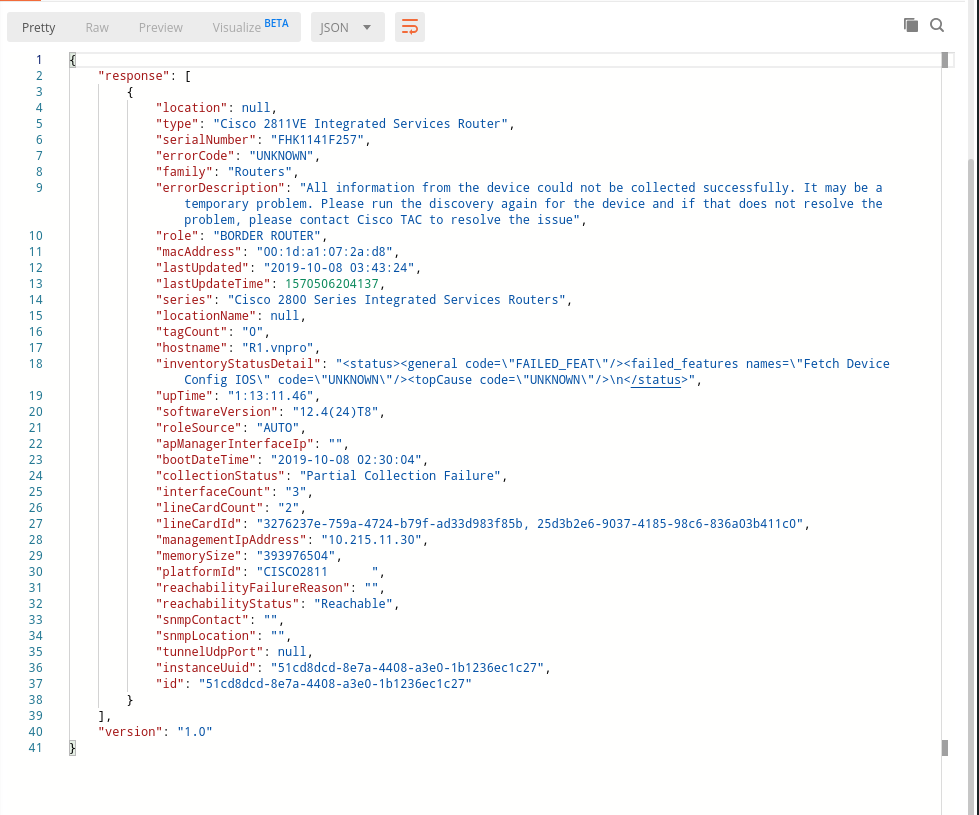
**Phần 2: Get list of device**

* Sau khi đã có Ticket, Tiến hành điền thêm headers cần thiết ở tab Headers

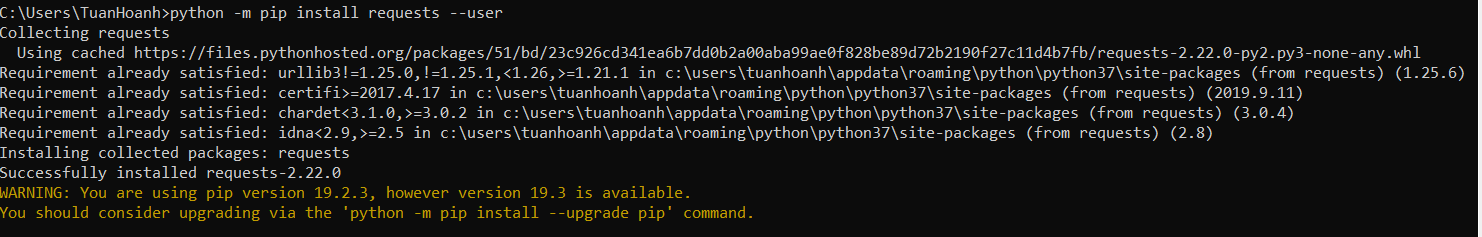


* Thêm Key “X-Auth-Token” với giá tri là Ticket vừa lấy được ở phía trên **[1]**
* **[2]** chuyển sang GET để lấy về list thiết bị
* **[3]** để lấy list of devicecần dùng REST function “/network-device”
* **[4]** nhấn send để APIC-EM trả về List of Devices

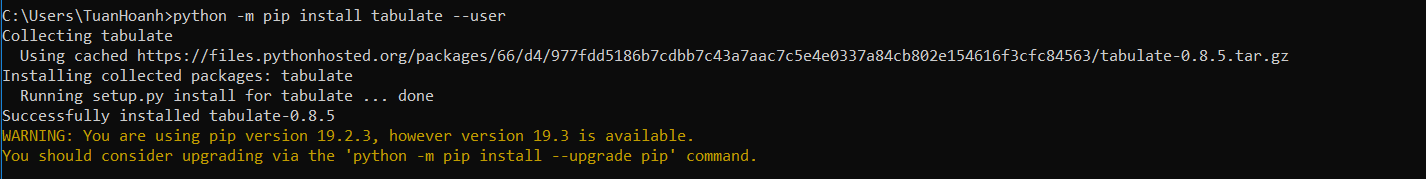
**Kết quả :**

**Viết chương trình bằng ngôn ngữ Python :**

Cài đặt thư viện requests :



Cài đặt thư viện tabulate :

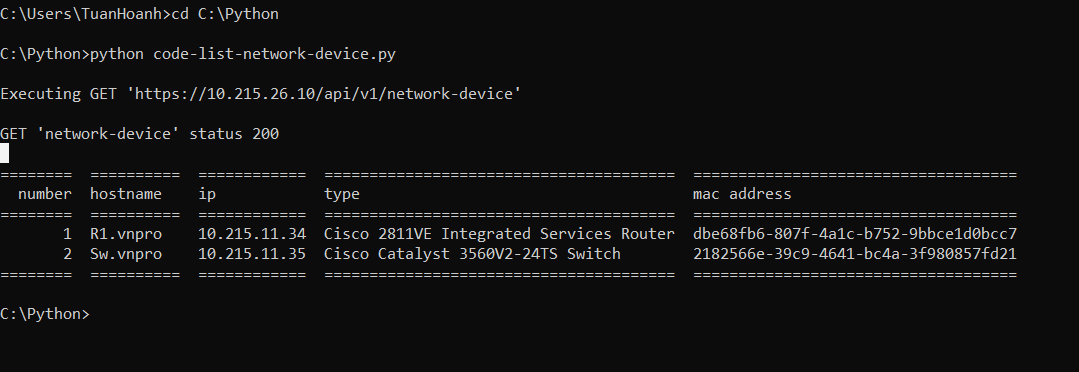


Vào phần mềm Atom, tạo new file apic-em.py và save C:\python.

|  |
| --- |
| import requests  import json  import sys  from tabulate import tabulate  APICEM\_IP = "10.215.26.10" *#Địa chỉ IP của APIC-EM*  USERNAME ="admin" *# Tên đăng nhập trong APIC-EM*  PASSWORD ="VnPro@149" *# Mật khẩu của tài khoản APIC-EM*  VERSION = "v1" *# Phiên bản APIC-EM version 1*  *#Tắt tin nhắn cảnh báo; requests là thư viện giúp gửi yêu cầu đến server và nhận response*  requests.packages.urllib3.disable\_warnings()  *'''Tạo hàm để lấy token;*  *các tham số ip,ver,uname,pword được gắn giá trị đã khai báo ở trên'''*  def get\_auth\_token(ip=APICEM\_IP,ver=VERSION,uname=USERNAME,pword=PASSWORD):  *#Tạo 1 jsonObject có 2 key username,password có value tương ứng là uname,pword*  r\_json = {  "username": uname,  "password": pword  }  post\_url = "https://"+ip+"/api/"+ver+"/ticket" #khai báo URL để post  headers = {"Content-Type" : "application/json"} #khai báo headers  *#try:...except: nghĩa là nếu code xảy ra lỗi ở trong phần try thì thực hiện phần except*  try:  r=requests.post(  post\_url,data = json.dumps(r\_json),  headers = headers,verify=False) *#gửi requests đến server và gán response vào biến r*  *#post là tạo dữ liệu ,post\_url là địa chỉ ta muốn gửi đến,*  *#json.dumps dùng để mã hóa username và password,*  *#verify=False để không xác thực SSL*  r.raise\_for\_status() *# Lấy mã trạng thái, 200 là thành công, 404 là not found*  token = r.json()["response"]["serviceTicket"] *#Lấy ticket từ r*  *# Trả về giá trị*  return {  "token" : token  }  except:  print("Status: %s" %r.status\_code)  print("Response: %s" %r.text)  *# Tạo hàm get để lấy thông tin*  def get(ip=APICEM\_IP,ver=VERSION,uname=USERNAME,pword=PASSWORD,api='',params=''):  ticket = get\_auth\_token() *#Lấy ticket bằng cách gọi hàm*  headers = {"X-Auth-Token": ticket['token']} *#Khai báo headers để gắn ticket vào*  url = "https://"+ip+"/api/"+ver+"/"+api  print("\nExecuting GET '%s'\n"%url) *#In ra thông tin để người dùng biết đang lấy dữ liệu*  try:  *#gửi request đến server, resp là lời đáp lại của yêu cầu, get là lấy dữ liệu*  resp = requests.get(url,headers=headers,params=params,verify=False)  print("GET '%s' status" %api,resp.status\_code,'\n') *#In ra trạng thái*  return(resp)  except:  print("Something wrong",api)  sys.exit()  device =[]  try:  resp = get(api="network-device") *# Lấy thông tin về network-device*  status = resp.status\_code *#Lấy trạng thái của yêu cầu trên*  response\_json = resp.json() *#Lấy nội dung json đã mã hóa từ lời đáp lại*  device = response\_json["response"] *#Gán thông tin từ lời đáp lại vào device*  except:  print("Something wrong,cannot get network device info")  sys.exit()  *# Nếu status khác 200(nghĩa là không thành công) thì in lời đáp lại và thoát chương trình*  if status !=200:  print (resp.text)  sys.exit()  *# Nếu chuỗi trống thi in ra không có thiết bị được tìm thấy và thoát chương trình*  if device == []:  print("No network device found")  sys.exit()  device\_list =[]  i=0  *#Tạo một vòng lặp để dò từng item và thực hiện gán các giá trị vào device\_list*  *#item["hostname"] là dò tìm key hostname trong item và lấy ra value*  for item in device:  i+=1  device\_list.append([i,item["hostname"],item["managementIpAddress"],  item["type"],item["instanceUuid"]])  *#thư viện tabulate dùng để khi in dữ liệu ra theo format đã được xây dựng sẵn*  print(tabulate(device\_list,  headers = ['number','hostname','ip','type','mac address'], tablefmt="rst")) |

Kiểm tra kết quả :

Vào cmd, thực hiện lệnh cd C:\python, sau đó chạy chương trình vừa viết bằng câu lệnh python apic-em.py :



Đây là chương trình lấy danh sách các thiết bị tự động nên nếu APIC-EM Controller có nhiều thiết bị sẽ lấy được thông tin thiết bị một cách nhanh chóng giúp tiết kiệm thời gian.