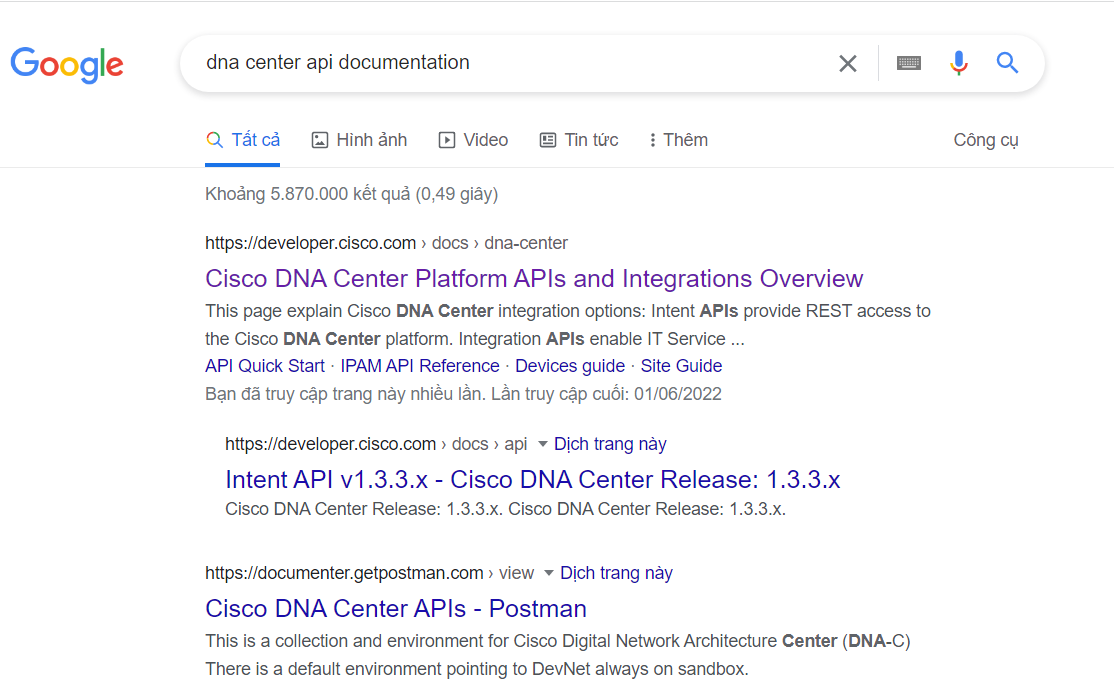
**Lab: 14: Sử dụng Python lấy danh sách thiết bị từ Network Controller (DNA CENTER) ( Cập nhật 15/06/2022)**

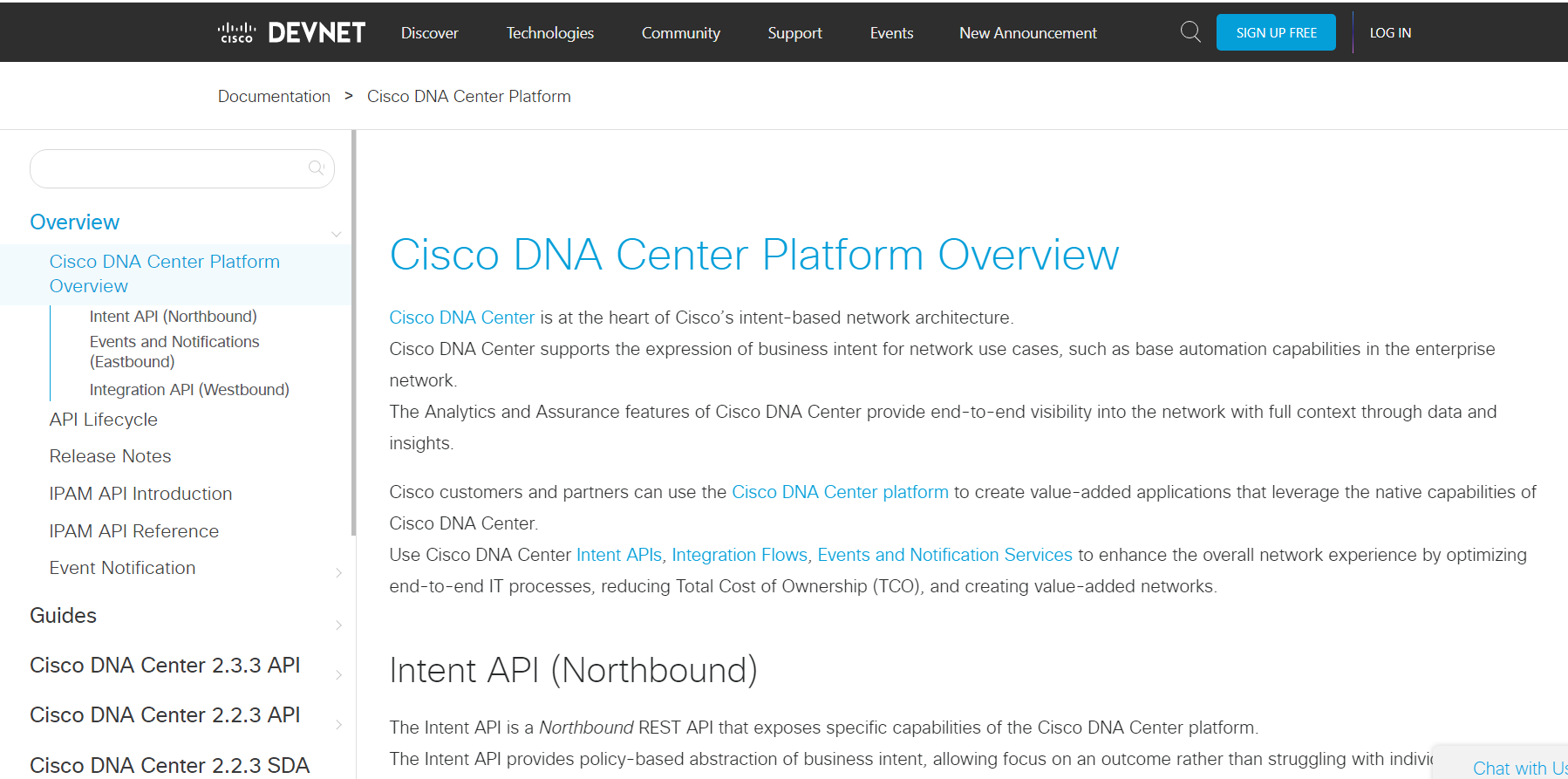
Có nhiều các tương tác với Network Controller thông qua các công cụ như Postman, Python, Curl,.. nhưng ngôn ngữ Python là một ngôn ngữ rất phổ biến trong thời gian gần đây cho phép chúng ta có thể tương tác với chúng và thực hiện các thao tác trên đó. Chính điều đó bài LAB hôm nay sẽ hướng dẫn mọi người thực hiện.

Đối với bất kỳ một Network Controller và DNA Center cũng không là ngoại lệ để thực hiện gọi API để lấy các danh sách thiết bị đầu tiên chúng ta phải thực hiện xác thực. Để thực hiện điều đó chúng ta truy cập đến <https://developer.cisco.com/docs/dna-center/> để xem các thông tin và thực hiện các bước để lấy thông tin từ chúng.

* Để xác thực đầu tiên chúng ta truy cập vào <https://developer.cisco.com/docs/dna-center/>

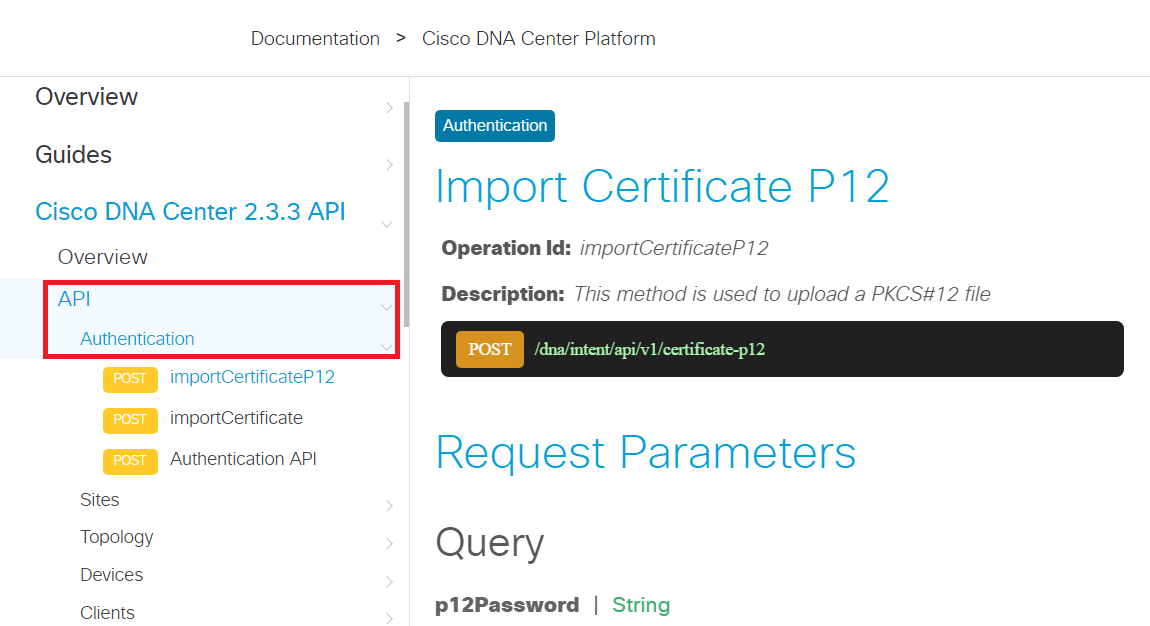


* Sau đó chọn **Cisco DNA Center Platform APIs and Integrations Overview**

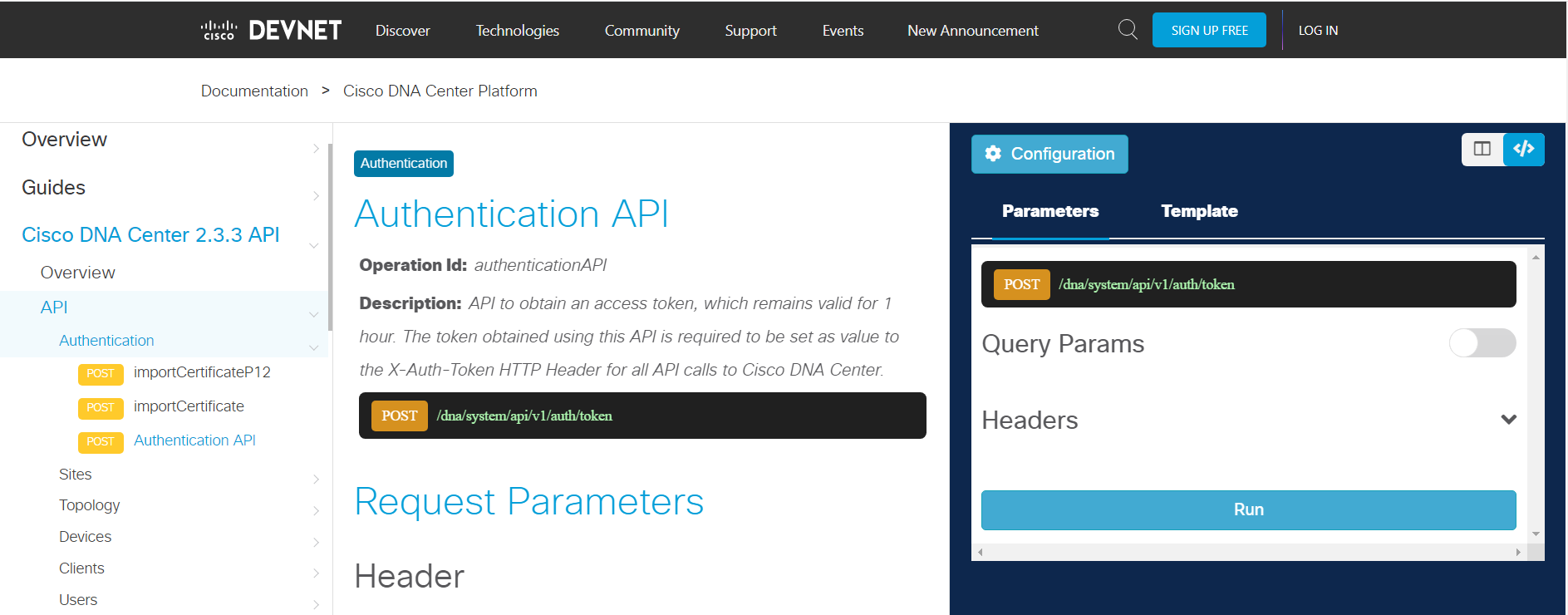


( Ở đây sẽ có các thông tin DNA Center bạn có thể tham khảo thêm cũng như là các phiên bản DNA Center được cập nhật. Mục đích chính của lab này là tương tác với API nên chúng ta sẽ cần quan tâm đến chúng)

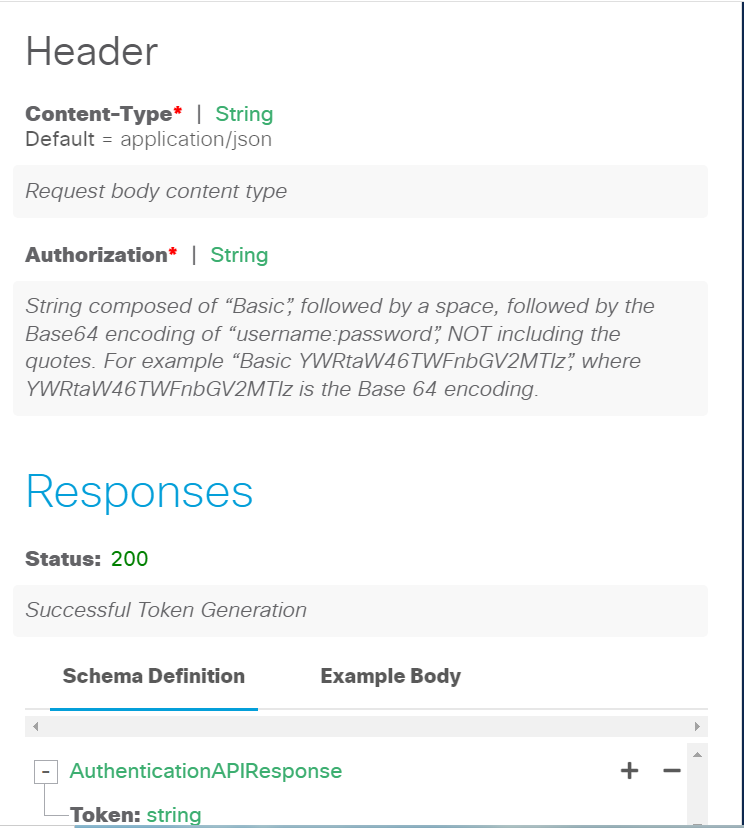
* Tiếp theo mình sẽ chọn phiên bản DNA Center mới nhất và chọn vào **API**



* Ở đây mình chọn **Authentication API** để xác thực:

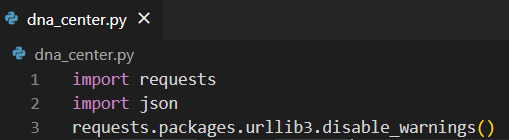


* Chúng ta có thể kéo xuống để xem thêm các thông tin:

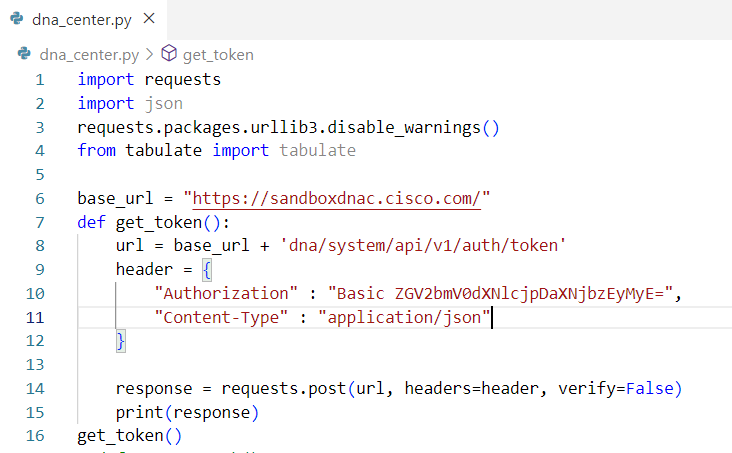


* Tiếp theo chúng ta sẽ mở công cụ thực hiện code và tạo một file mới .py để thực hiện:

+ Import các thư viện cần thiết ( sử dụng thư viện requests và json để format dữ liệu và tắt cảnh báo SSL):



+ Đầu tiên chúng ta sẽ tạo một hàm để trả về token khi xác thực thành công:



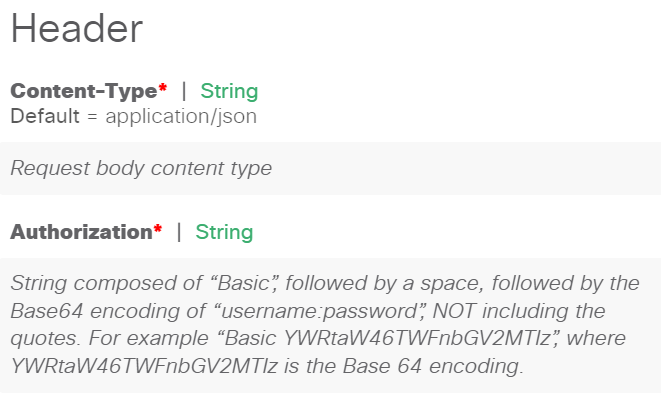
+ Ghi chú:

base\_url: đường dẫn cơ sở ( ở đây mình sử dụng sanbox của Cisco)

url: chúng ta có thể xem thông tin trên Document

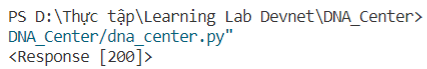


header: chúng ta xem tại Document



* Đối với Value trong Authorization thì chúng đã mã hóa username và password theo base 64 encode ( Ở đây mình sử dụng username, password có sẳn trên Devnet Learning Labs bạn có thể tham khảo tại đây: <https://developer.cisco.com/learning/modules/dnac-rest-apis/dnac-101-auth/get-started-with-authentication-api-postman/> )

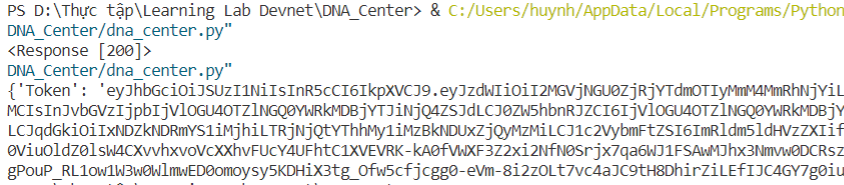
response: kết quả trả về chúng ta sử dụng thư viện requests và điền các biến thích hợp vào bên trên và in ra màn hình



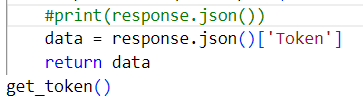
* Kết quả trả về thành công

+ Nhưng để xem kết quả trả về chúng ta sử dụng thư viện json để định dạng kết quả trả về và in ra màn hình:



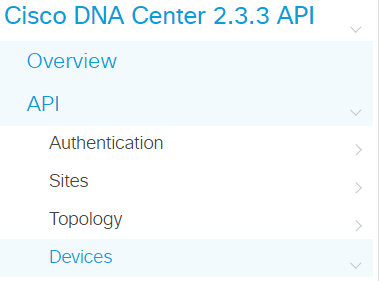


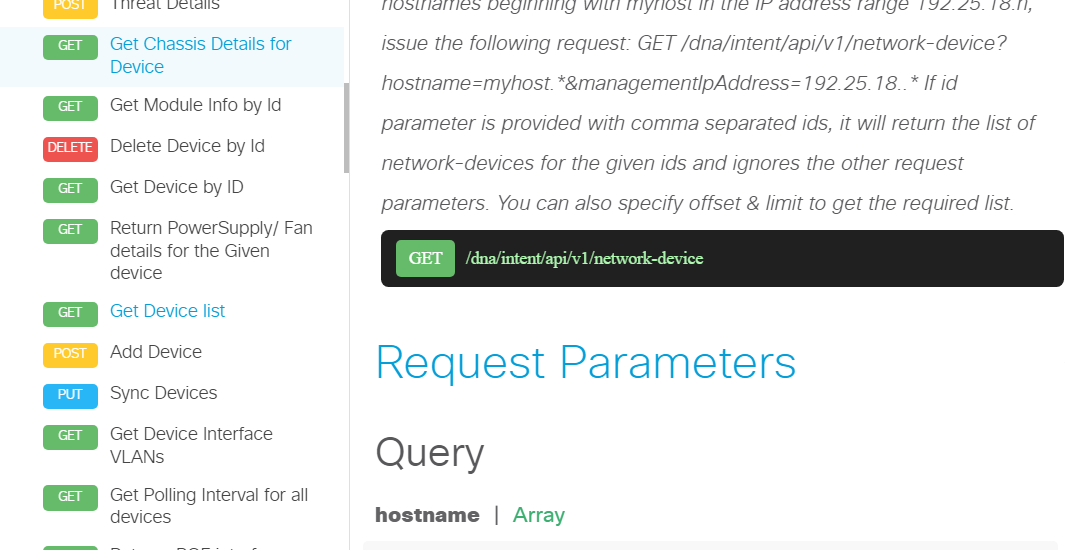
+ Sau đó chúng ta thực hiện trích xuất phần tử bên trong kết quả trả về là Token và return kết quả trả về:



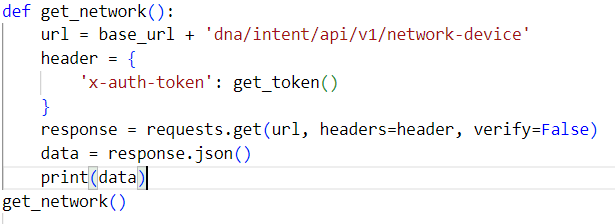
+ Tiếp theo chúng ta thực hiện lấy danh sách thiết bị sau khi đã xác thực thành công. Chúng ta cũng sẽ viết thêm một hàm để thực hiện điều đó. Đầu tiên để lấy được thiết bị chúng ta cần vào API Document để xem thông tin:

+ Vào **Device** và tìm **GetDeviceList:**





+ Tương tự đối với xác thực chúng ta thực hiện tạo các biến và điền các thông tin thích hợp do Document cung cấp ( và gọi hàm đã xác thực ở trên để lấy token) :



+ Sau đó chúng ta thực hiện in kết quả ra màn hình bằng cách sử dụng thư viện json() để định dạng chúng



+ Kết quả trả về danh sách các thiết bị nhưng không theo định dạng nên khó phân biệt vì vậy thư viện json() cũng hỗ hợ cách để chuyển về định dạng dễ phân biệt hơn:



json.dumps() cho phép định dạng lại với thông số thích hợp ở đây mình chọn là 4 ( khoảng cách giữa các tab)



🡪Như vậy chúng ta đã có thể thực hiện lấy danh sách các thiết bị từ DNA Center thông qua Python

Nội dung file code:

import requests

import json

requests.packages.urllib3.disable\_warnings()

base\_url = "https://sandboxdnac.cisco.com/"

def get\_token():

    url = base\_url + 'dna/system/api/v1/auth/token'

    header = {

        "Authorization" : "Basic ZGV2bmV0dXNlcjpDaXNjbzEyMyE=",

        "Content-Type" : "application/json"

    }

    response = requests.post(url, headers=header, verify=False)

    #print(response.json())

    data = response.json()['Token']

    return data

get\_token()

def get\_network():

    url = base\_url + 'dna/intent/api/v1/network-device'

    header = {

        'x-auth-token': get\_token()

    }

    response = requests.get(url, headers=header, verify=False)

    data = response.json()

    print(json.dumps(data, indent=4))

get\_network()