

## Programación de Redes

# 2.5 Lab - RESTCONF

**PRESENTA:**

**NOMBRE: ROSAS MUÑOZ ANGEL DE JESUS**

**NUMERO DE CONTROL: 1219100391**

**NUMERO DE LISTA: #12**

**GRUPO: GIR0441**

**PROF: Gabriel Barrón Rodríguez**

ISU en  
**Infraestructura de  
Redes Digitales**

Dolores Hidalgo, Gto. Septiembre-diciembre

### **¿Que es el modelo RESTCONF?**

El modelo Restconf es un protocolo de http, este nos ayuda a las configuraciones rest, algo así como netconf

Este modelo envía los datos por medio de la encapsulación HTTP.

### **¿Como lo utilizo con Python?**

Para esta practica necesitamos el link o api que anteriormente generamos en Postman

Después solo utilizamos la función requests que es con la cual pudimos solicitar los datos de ese link, a su vez utilizamos la librería de JSON para parpear los demás datos que son los headers y la configuración, dejarlo en una variable y mandarla a imprimir.

### **¿Qué configuración tendría que realizar?**

Para este caso no se necesitó mucha configuración ni código solo con el link y los parámetros de los headers para solicitar el REST

### **¿El tipo de formatos que se manejan?**

JSON y XML aun que se utiliza mas JSON

```
lab 2.5.py - D:/PYTHON/lab 2.5.py (3.10.7)
File Edit Format Run Options Window Help

import json
import requests
requests.packages.urllib3.disable_warnings()

api_url = "https://10.10.20.48/restconf/data/ietf-interfaces:interfaces"

headers = { "Accept": "application/yang-data+json",
            "Content-type": "application/yang-data+json"
          }

basicauth = ("developer", "Cisco12345")

resp = requests.get(api_url, auth=basicauth, headers=headers, verify=False)

response_json = resp.json()

print(response_json)

IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help

Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64
AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/PYTHON/lab 2.5.py =====
>>>
===== RESTART: D:/PYTHON/lab 2.5.py =====
{'ietf-interfaces:interfaces': {'interface': [{'name': 'GigabitEthernet1', 'description': 'MANAGEMENT INTERFACE - DON'T TOUCH ME', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '10.10.20.48', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}, {'name': 'GigabitEthernet2', 'description': 'Network Interface', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': False, 'ietf-ip:ipv4': {}, 'ietf-ip:ipv6': {}}, {'name': 'GigabitEthernet3', 'description': 'Network Interface', 'type': 'iana-if-type:ethernetCsmacd', 'enabled': False, 'ietf-ip:ipv4': {}, 'ietf-ip:ipv6': {}}, {'name': 'Loopback1', 'type': 'iana-if-type:softwareLoopback', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '1.1.1.1', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}, {'name': 'Loopback99', 'type': 'iana-if-type:softwareLoopback', 'enabled': True, 'ietf-ip:ipv4': {'address': [{'ip': '2.2.2.2', 'netmask': '255.255.255.0'}]}, 'ietf-ip:ipv6': {}}]}}
```

Símbolo del sistema

Vínculo: dirección IPv6 local. . . : fe80::9c08:d3ef:3do)

Dirección IPv4. . . . . : 192.168.1.1

Máscara de subred. . . . . : 255.255.255.0

Concesión obtenida. . . . . : miércoles, 7 de 2022 07:00:11 p. m.

La concesión expira. . . . . : jueves, 11 de 2022 08:09:51 p. m.

Puerta de enlace predeterminada. . . . : fe80::1%4192.168.1.254

Servidor DHCP. . . . . : 192.168.1.1

IAID DHCPv6. . . . . : 53767972

DUID de cliente DHCPv6. . . . . : 00-01-00-00-5F-24-F2-CB-DF

Servidores DNS. . . . . : 2806:1020:2806:1030:ffff:103192.168.1.254

NetBIOS sobre TCP/IP. . . . . : habilitado

Adaptador de Ethernet Conexión de red Bluetooth:

Estado de los medios. . . . . : medios des

Sufijo DNS específico para la conexión. . :

Descripción. . . . . : Bluetooth (Area Network)

Dirección física. . . . . : 34-6F-24-F

DHCP habilitado. . . . . : sí

Configuración automática habilitada. . . : sí

```
lab 2.5par2.py - D:\lab 2.5par2.py (3.10.7)
File Edit Format Run Options Window Help
import json
import requests
requests.packages.urllib3.disable_warnings()

api_url = "https://10.10.20.48/restconf/data/ietf-interfaces:interfaces/interface

headers = {
    "Accept": "application/yang-data+json",
    "Content-type": "application/yang-data+json"
}
basicauth = ("developer", "Cisco12345")

yangConfig = {
    "ietf-interfaces:interface": {
        "name": "Loopback99",
        "description": "Laboratorio 2.5 prueba",
        "type": "iana-if-type:softwareLoopback",
        "enabled": True,
        "ietf-ip:ipv4": {
            "address": {
                "ip": "99.99.99.99",
                "netmask": "255.255.255.0"
            }
        },
        "ietf-ip:ipv6": {}
    }
}

resp = requests.put(api_url, data=json.dumps(yangConfig), auth=basicauth, header
if (resp.status_code >= 200 and resp.status_code <= 299):
    print("STATUS OK: {}".format(resp.status_code))
else:
    print("Error code {}, reply: {}".format(resp.status_code, resp.json()))
```

```
lab 2.5.py - Unidad_3 - Visual Studio Code
Símbolo del sistema
Vínculo: dirección IPv6 local. . . : fe80::9c08:d3ef:34a1:5db4%4(Preferido)
Dirección IPv4. . . . . : 192.168.1.204(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Concesión obtenida. . . . . : miércoles, 30 de noviembre de 2022 07:00:11 p. m.
La concesión expira . . . . . : jueves, 1 de diciembre de 2022 08:09:51 p. m.
Puerta de enlace predeterminada . . . . : fe80::1%4
192.168.1.254
Servidor DHCP . . . . . : 192.168.1.254
IAID DHCPv6 . . . . . : 53767972
DUID de cliente DHCPv6. . . . . : 00-01-00-01-2A-B2-93-A2-34-6F-24-F2-CB-DF
Servidores DNS. . . . . : 2806:1020:ffff:103::e
2806:1030:ffff:103::e
192.168.1.254
NetBIOS sobre TCP/IP. . . . . : habilitado

Adaptador de Ethernet Conexión de red Bluetooth:
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :
Descripción . . . . . : Bluetooth Device (Personal Area Network)
Dirección física. . . . . : 34-6F-24-F2-CB-DE
DHCP habilitado . . . . . : sí
Configuración automática habilitada . . : sí
```