



UNIVERSIDAD TECNOLÓGICA DEL NORTE DE GUANAJUATO

Tecnologías de la Información y Comunicación
Programa educativo:

TSU en Infraestructura de Redes Digitales
Área académica:

Programación de Redes
Asignatura:

Unidad III: Programación de Redes

Grupo: GIR0441

RESTCONF with Postman
Laboratorio 4:

Venado Soria German Emiliano
Alumno:

Gabriel Barrón Rodríguez
Docente:

Dolores Hidalgo, C.I.N., Gto., Miércoles 14 de Diciembre de 2022
Lugar y fecha:

Lab – RESTCONF with Postman

Objectives

Part 1: Setup HTTP Headers in Postman

Background / Scenario

RESTCONF is a RESTful API interface that provides a programmatic interface for accessing data defined in YANG device models.

In this lab, you will learn how to interact with the RESTCONF interface using the Postman application to retrieve the device's configuration, update and create new interfaces and retrieve operation data.

Required Resources

- Postman
- Access to a router with the IOS XE operating system version 16.6 or higher.

Instructions

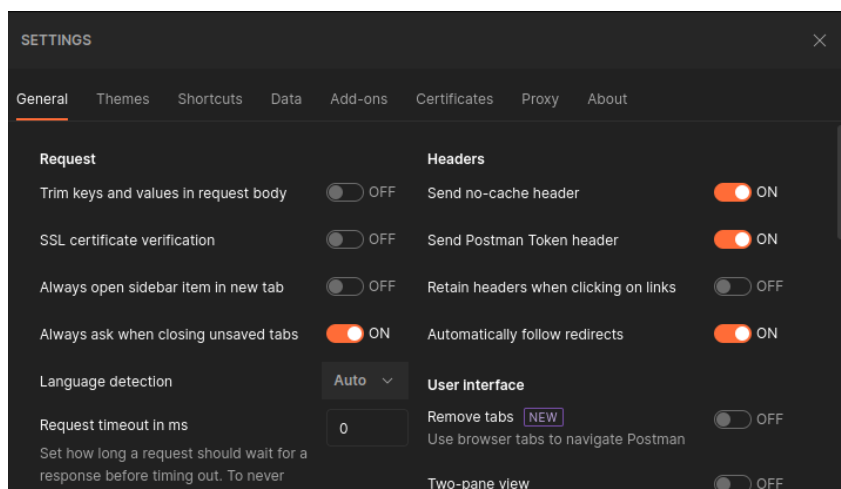
Part 1: Setup HTTP headers and Authentication in Postman

In this part, you will setup the HTTP Headers in Postman to work with RESTCONF REST API interfaces.

Step 1: Configure Postman.

Configure a Postman setting.

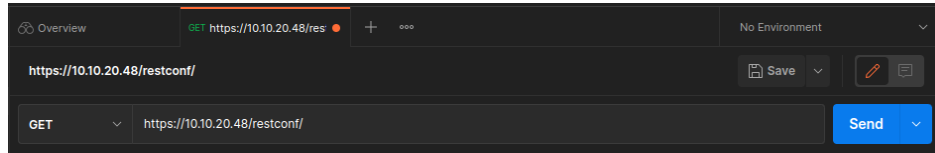
- Open Postman and create a new tab if necessary.
- Click **File > Settings**.
- Under the **General** tab, set the **SSL certificate verification** to **OFF**. Close the **Settings** dialog box.



En este primer paso se requiere de interactuar con POSTMAN, y nos pide que primeramente nos vayamos a la configuración para desactivar la verificación del certificado SSL.

Step 2: Select the method and enter the required URL.

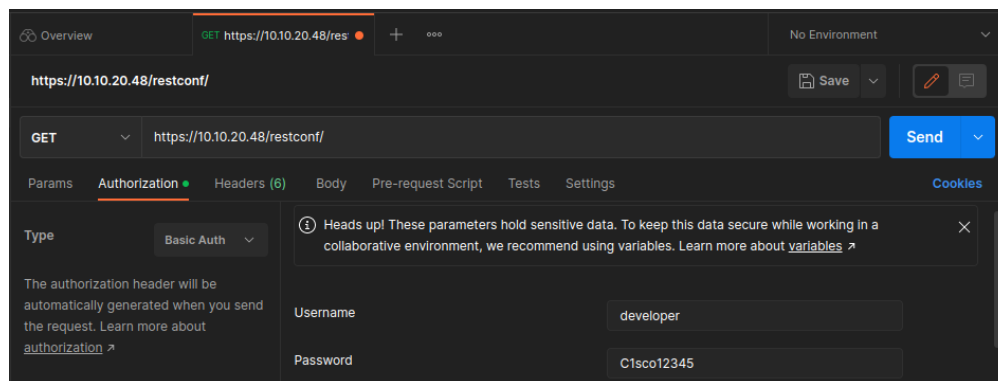
- Next to the URL box, select the request method to be **GET**.
- Enter the URL for API endpoint: <https://192.168.56.101/restconf/> (adjust the IP address to match the router's current address)



Como segundo paso, nos indica que realicemos una solicitud GET, donde debemos ingresar la dirección del Router que hemos estado trabajando constantemente.

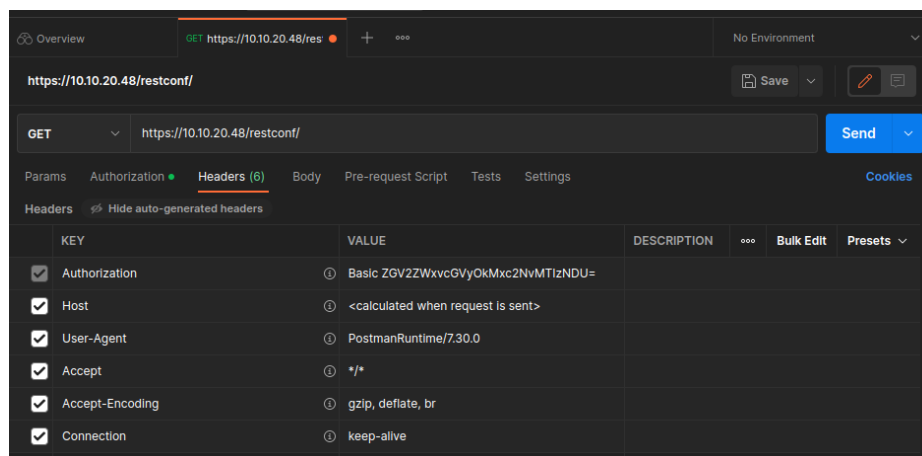
Step 3: Enter the authentication header information.

- Below the URL input field, select **Authorization**.
- Select the type **Basic Auth** and enter the username and password for authenticating to the RESTCONF API service:
 developer
 C1sco12345



Posteriormente, se solicita hacer la autenticación básica donde estaremos logeando el usuario y contraseña que tiene el Router por predeterminado.

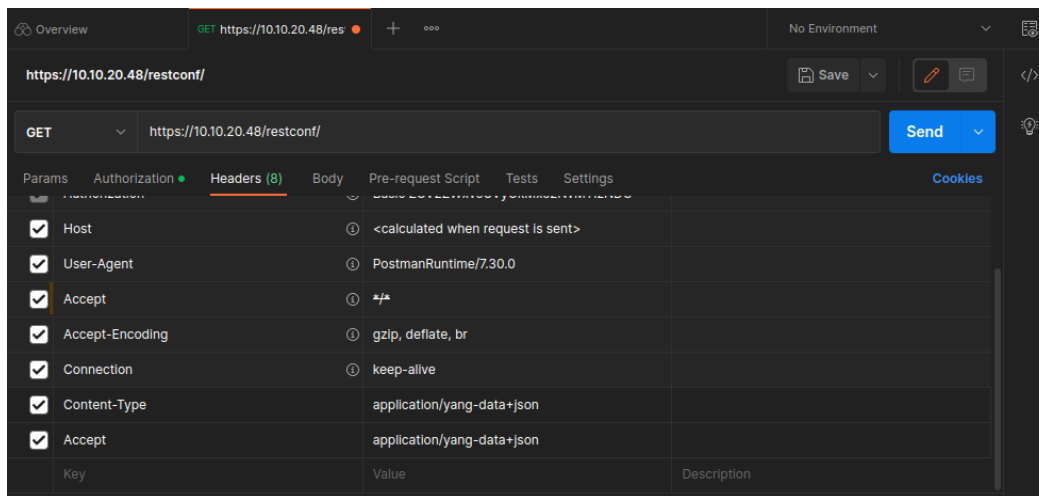
- Click on the **Preview Request** to add the Authentication header to the Headers.



Damos clic en “Headers” para visualizar los encabezados que se generaron predeterminadamente.

Step 4: Enter the additional Headers.

- Next, below the URL input field, select **Headers**.
- Under **Headers**, click the **New key** field under the **Key** column and enter **Content-Type** to define what is the encoding of data that are sent with Postman in the REST API request.
- Next again under **Headers**, click the **New key** field under the **Key** column and enter **Accept** to define in what encoding you wish to receive back the data replies from the REST API request.
- In the value columns, enter `application/yang-data+json` when sending or receiving data that are in JSON encoding. When sending or receiving data in XML encoding (default), use `application/yang-data+xml`.

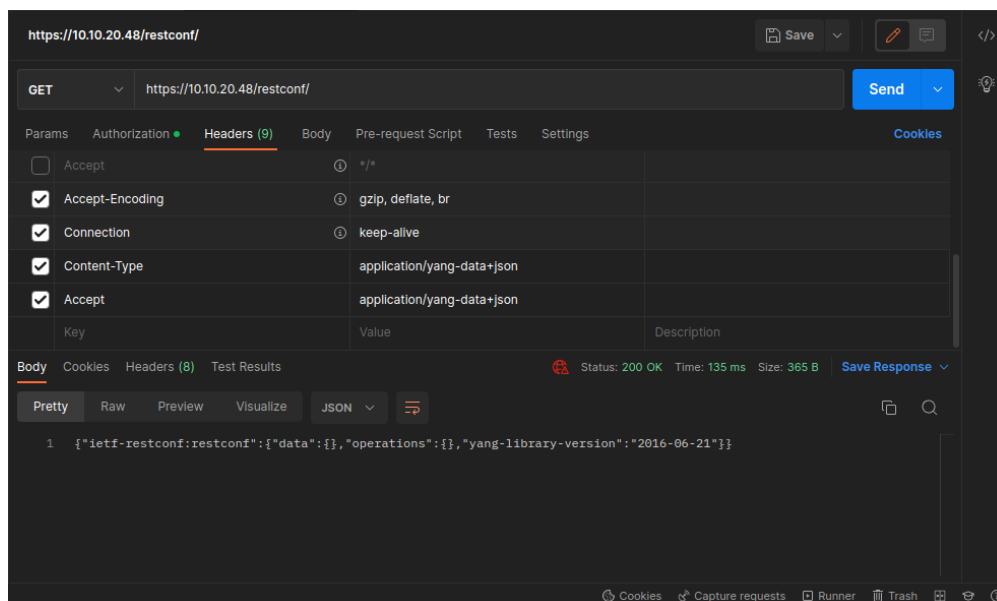


En este siguiente paso, se agregan dos nuevos encabezados llamados “Content-Type” y “Accept” en el cual vamos a interactuar con dos tipos de valores, de tipo json y xml.

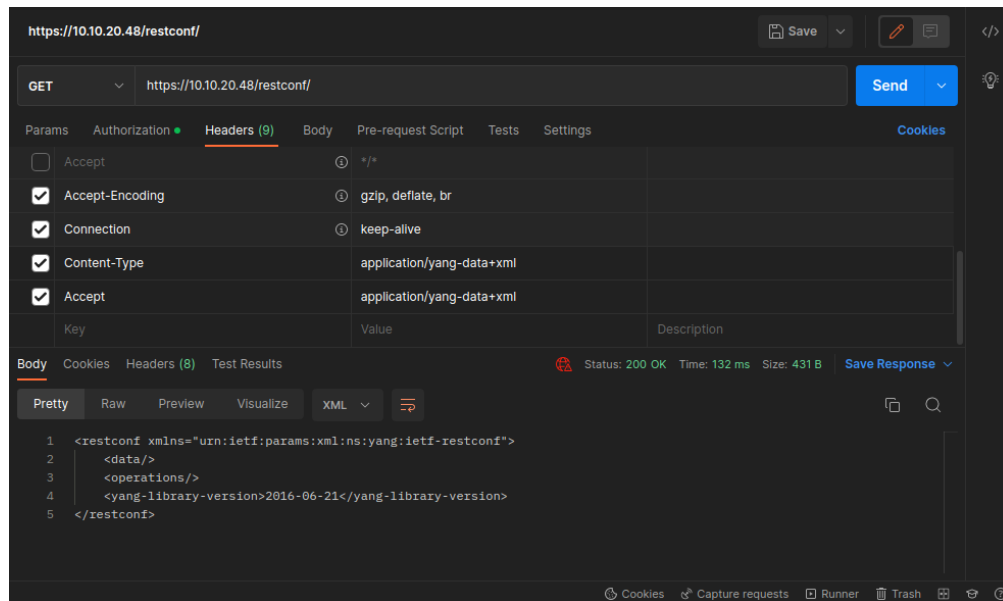
Step 5: Send the request.

- Click **Send**.

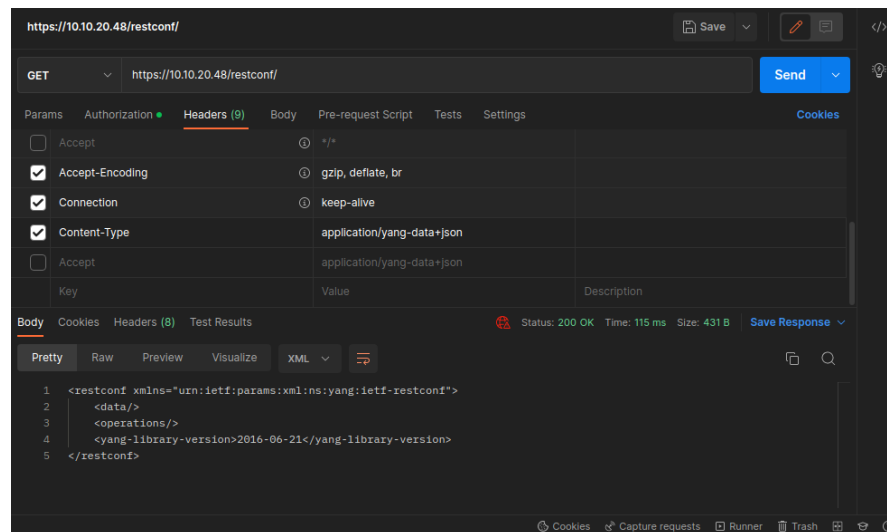
Salida con valor de: `application/yang-data+json`



Salida con valor de: application/yang-data+xml



- b. After a brief delay, you should see the response appear below the request information. Scroll down, if necessary to view the response data.
- c. When the **Accept** header was set to `application/yang-data+json` the response consists of JSON encoded data.
- d. Uncheck the **Accept** header, or change it to `application/yang-data+xml` (default behavior when no Accept header is defined) and click on **Send**. Now the response consists of XML encoded data:



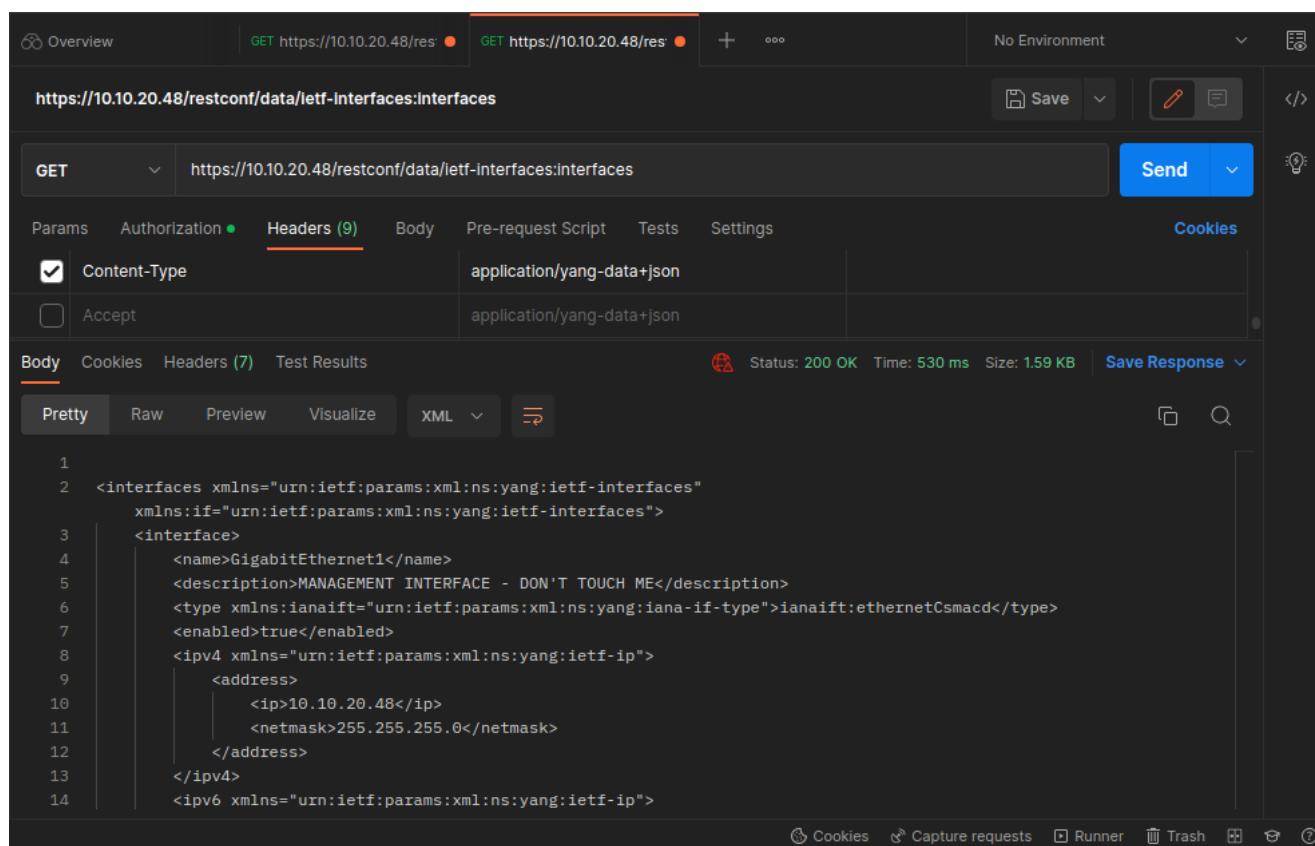
Cuando desactivamos el encabezado de “Accept” o agregamos solamente el valor de “application/yang-data+xml” a ese encabezado, la salida automática será en formato xml, tal y como se muestra en la captura.

Note: If there is an error, check the Status of the request. Check the meaning of the status code. 200 means success. A 404 error may mean that the URL was entered incorrectly. A 401 or 403 error could indicate a problem with the authentication, so check that the credentials are entered correctly.

Part 2: Retrieve Interface Configuration

Step 1: Retrieve YANG data in XML.

- Duplicate the existing REST API request in Postman by right clicking on the request and selecting Duplicate Tab to create a new request with prefilled existing Headers:
- In the URL for the API endpoint, enter after the host and port portion:
`/restconf/data/ietf-interfaces:interfaces` to retrieve data defined in the **ietf-interfaces** YANG model in the **interfaces** container.
- Click **Send** to send the RESTCONF API request. With unchecked **Accept** header or the **Accept** header set to `application/yang-data+xml`, the result YANG data are returned in XML encoding.
- Check the Accept header and set it to `application/yang-data+json` to receive YANG data formatted in JSON encoding.



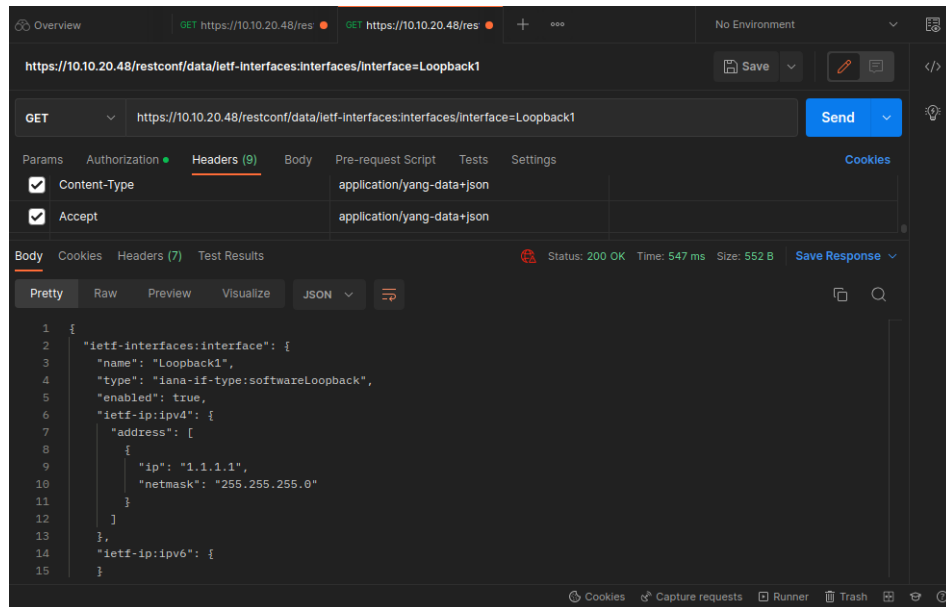
En este siguiente paso se menciona que se debe hacer una duplicación de la solicitud API REST para poder agregar otros parámetros a la copia. En las instrucciones nos dice como es que se realiza una copia, simplemente con un clic derecho y seleccionar la opción correspondiente.

Y en esta se va a modificar la URL, agregando `/restconf/data/ietf-interfaces:interfaces` al final de la URL anterior, y esto nos muestra datos del Router en modelo YANG pero en formato xml, esta información se encuentra en el contenedor de interfaces.

Step 2: Retrieve information about a single interface

- In the URL for the API endpoint, enter after the host and port portion:
`/restconf/data/ietf-interfaces:interfaces/interface=Loopback1` to retrieve data for the **Loopback1** interface only.

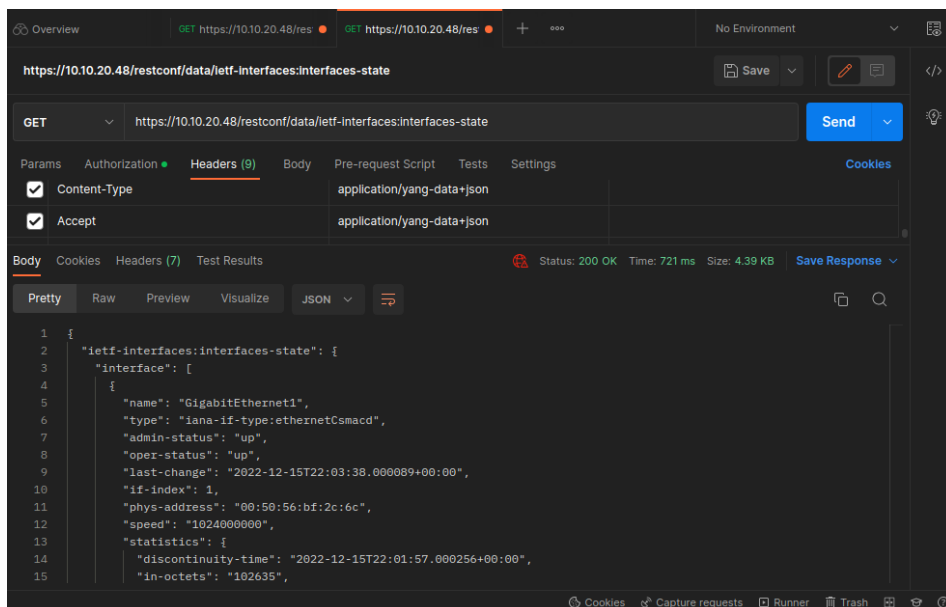
- b. Click **Send** ...



En este apartado se pide nuevamente modificar la URL, esto para solicitar información solamente de la interfaz Loopback1.

Step 3: Retrieve operation data

- a. Change the RESTCONF API Endpoint to retrieve operation and statistical data about the interfaces. You can use the “pyang” tool to discover what container in the **ietf-interfaces** YANG model includes the operational and stats data.



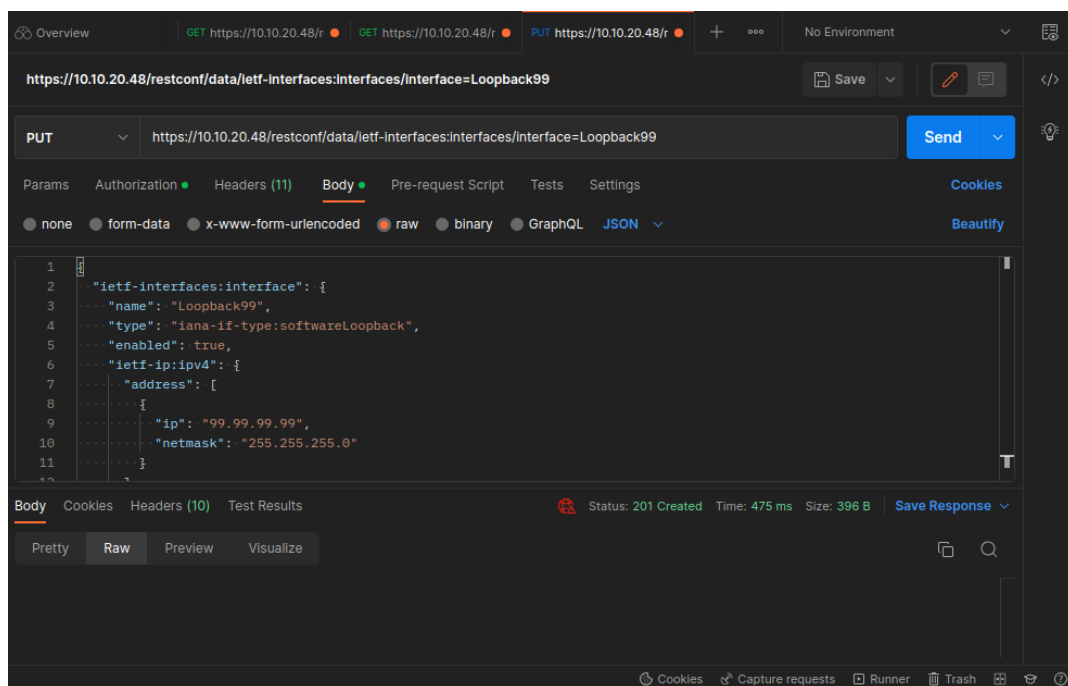
Para recuperar datos estadísticos y de operación sobre las interfaces se agrega al final “state”.

- b. What is the new URL?

R= La nueva URL es `https://10.10.20.48/restconf/data/ietf-interfaces:interfaces-state`

Part 3: Update Interface Configuration

- Duplicate the existing REST API request in Postman by right clicking on the request and selecting Duplicate Tab to create a new request with prefilled existing Headers.
- In the URL for the API endpoint, change the method from **GET** to **PUT** and enter after the host and port portion: `/restconf/data/ietf-interfaces:interfaces/interface=Loopback99` to create a new interface **Loopback99**.
- In the Body part of the request, enter the YANG data you want to apply to the device's configuration (copy, paste and customize the JSON data from the previous GET request on Loopback1 – change the interface name and IP address).
- Click **Send** to create a new interface Loopback99 with the IP address 99.99.99.99/24.
- In the Output (scroll down), you should see the HTTP Error code **201 Created** indicating that the new interface has been created:

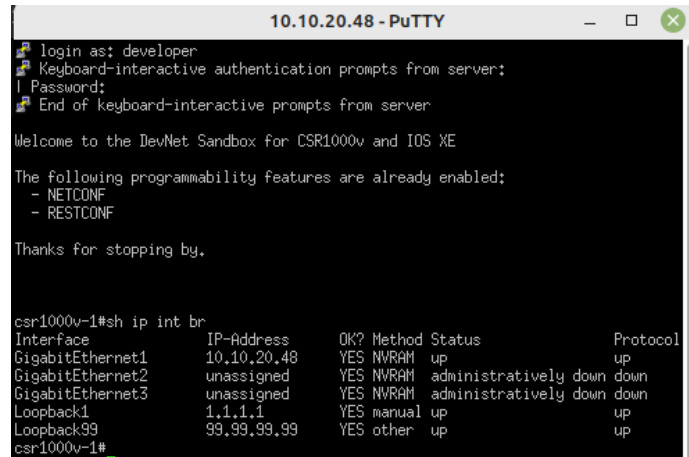


Otra vez se repite el paso de duplicar la solicitud API REST, pero en esta ocasión la solicitud será con el método PUT, esto para que se pueda enviar código hacia el Router. La información que estaremos enviando es para crear una segunda interfaz Loopback con el nombre “99”.

Para facilitar el proceso nos sugiere copiar el cuerpo de los datos YANG de la interfaz loopback1, ya que es la misma estructura para crearla y la pegamos en la opción “raw” del apartado de “Body”, solamente cambiamos el nombre y dirección IP de la interfaz nueva

Y como podemos apreciar, cuando envié mi script, me arrojó el mensaje de 201, el cual nos indica que se ha creado correctamente nuestra interfaz.

- Verify using the IOS CLI that the new Loopback99 interface has been created (sh ip int brief).



```
10.10.20.48 - PuTTY
login as: developer
Keyboard-interactive authentication prompts from server:
Password:
End of keyboard-interactive prompts from server

Welcome to the DevNet Sandbox for CSR1000v and IOS XE

The following programmability features are already enabled:
- NETCONF
- RESTCONF

Thanks for stopping by.

csr1000v-1#sh ip int br
Interface      IP-Address      OK? Method Status        Protocol
GigabitEthernet1  10.10.20.48    YES NVRAM  up            up
GigabitEthernet2  unassigned     YES NVRAM  administratively down down
GigabitEthernet3  unassigned     YES NVRAM  administratively down down
Loopback1        1.1.1.1        YES manual up            up
Loopback99       99.99.99.99    YES other  up            up
csr1000v-1#
```

Además, nos pide verificar en el Router con el siguiente comando: **show ip interface brief**, y como podemos apreciar, ahí aparece la segunda interfaz que fue creada con el nombre 99, además indican su respectiva dirección IP.

- g. Update the RESTCONF API request to delete the interface **Loopback99**.
- h. What changes were needed to be made in the RESTCONF request to delete the interface?

Investigaciones

• ¿Qué es POSTMAN?

Es una aplicación que nos permite realizar pruebas API. Es un cliente HTTP que nos da la posibilidad de testear 'HTTP requests' a través de una interfaz gráfica de usuario, por medio de la cual obtendremos diferentes tipos de respuesta que posteriormente deberán ser validados.

• Métodos

Postman nos ofrece muchos métodos para interactuar con los 'endpoints'. Los más utilizados y sus funciones son:

- GET: Obtener información
- POST: Agregar información
- PUT: Reemplazar la información
- PATCH: Actualizar alguna información
- DELETE: Borrar información

• Encabezados preestablecidos

Cuando desea interactuar con una API, se requieren algunos encabezados cada vez que realiza una solicitud. Los encabezados incluyen nombre de usuario, contraseña, clave de API, autorización, etc., pero cuando trabaja con la aplicación, se configura automáticamente y envía la solicitud. En caso de acceder directamente a la API, debe pasar esos encabezados cada vez que necesite realizar una solicitud.

Conclusiones

De este laboratorio me llevo una gran experiencia, la primera vez que trate de trabajar con POSTMAN tuve muchas dificultades, pues carecía de muchos conocimientos, pero gracias a esta práctica a algunas investigaciones que realice, pude interactuar sin ningún problema.

Se me hizo similar cuando trabajamos con las API's, pues son solicitudes de respuesta con diferentes métodos, al principio tuve un poco de confusión con los métodos, pero al final comprendí que función hacia cada uno, simplemente trabajamos con dos, pero quede impresionado con el PUT, pues este nos ayudo a modificar datos para generar una interfaz de manera remota estando en POSTMAN.