



Universidad Tecnológica
del Norte de Guanajuato
Organismo Público Descentralizado del Gobierno del Estado

"Educación y progreso para la vida"

Tecnologías de la Información y Comunicación

Programa educativo: TSU en Infraestructura de Redes Digitales

Área académica: Programación de Redes

Asignatura:

Unidad 3

Grupo: GIR0441

2.4 Lab – RESTCONF with Postman.docx

Alumna Gómez Luna Cinthia Valeria:

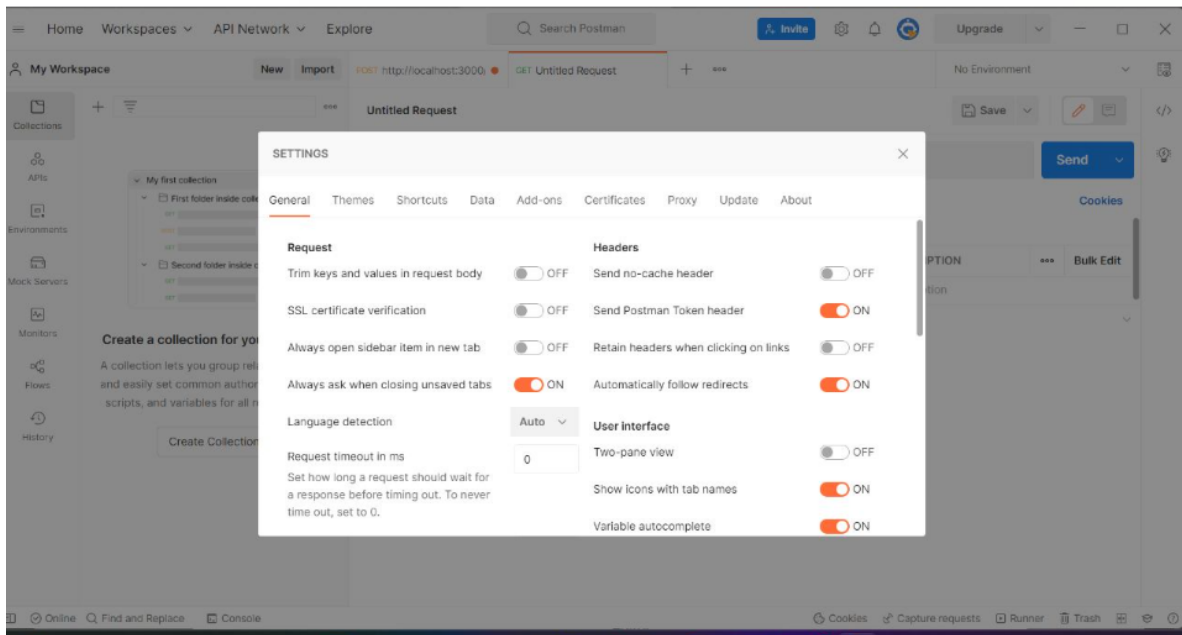
Docente: Gabriel Barrón Rodríguez

Dolores Hidalgo, C.I.N., Gto., jueves 24 de noviembre de 2022

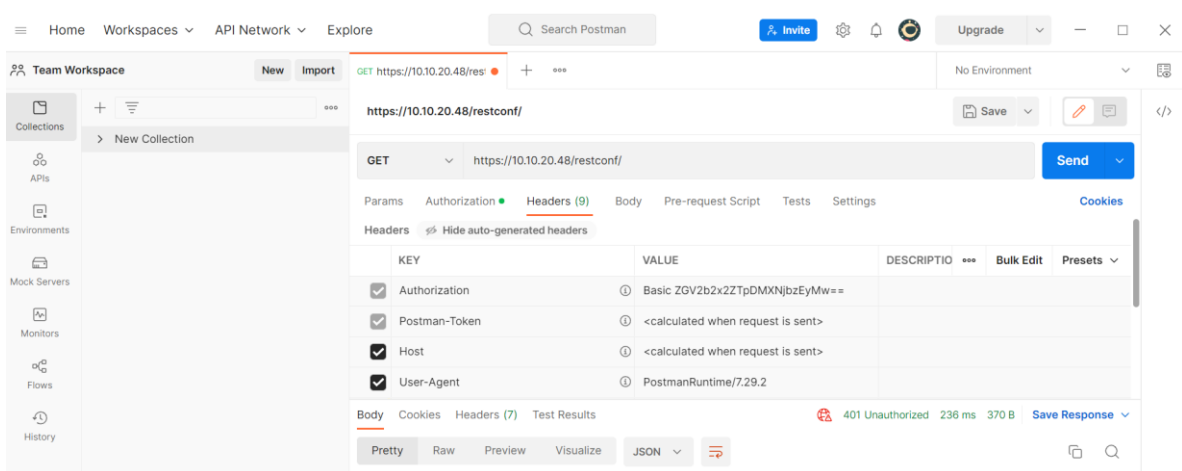
Introducción

Postman es una aplicación que nos permite testear APIs a través de una interfaz gráfica de usuario. Entre las ventajas que tiene Postman encontramos la capacidad de crear colecciones y distintos ambientes de pruebas. Postman es una herramienta fácil de usar que nos ayuda a optimizar el tiempo de ejecución de pruebas. Se dice que su lengua que utiliza está apoyada en código de Javascript para que así programen su tests, en el siguiente paso se mostraran unas capturas de pantalla y una breve explicación de lo que se logró hacer.

Configuración de Postman



Ingresando URL



additional Headers

The screenshot shows the Postman interface with a GET request to `https://10.10.20.48/restconf/`. The 'Headers' tab is selected, showing 10 headers:

| Key | Value | Description |
|-----------------|----------------------------|-------------|
| Accept-Encoding | gzip, deflate, br | |
| Connection | keep-alive | |
| Content-Type | application/yang-data+json | |
| Accept | application/yang-data+json | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

The status bar at the bottom indicates a 401 Unauthorized response with 236 ms latency and 370 B body size.

SEND_GET

The screenshot shows the Postman interface with a GET request to `https://10.10.20.48/restconf/`. The 'Authorization' tab is selected, showing Basic Authentication with the username 'developer' and password 'C1sco12345'. The 'Body' tab is also visible, showing a JSON response:

```
1 {
2   "ietf-restconf:restconf": {
3     "data": {},
4     "operations": {},
5     "yang-library-version": "2016-06-21"
6   }
7 }
```

The status bar at the bottom indicates a 200 OK response with 424 ms latency and 365 B body size.

SEND_XML

The screenshot shows the Postman interface with a REST client request configured. The URL is `https://10.10.20.48/restconf/`. The method is GET. The headers tab is selected, showing 10 headers. The body tab is also selected, showing an XML payload. The status bar indicates a 200 OK response with a 264 ms latency and 431 B of data.

Headers (10)

| Key | Value | Description |
|--------------|---------------------------|-------------|
| Connection | keep-alive | |
| Content-Type | application/yang-data+xml | |
| Accept | application/yang-data+xml | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Body

XML

```
1 <restconf xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf">
2   <data/>
3   <operations/>
4   <yang-library-version>2016-06-21</yang-library-version>
5 </restconf>
```

200 OK 264 s 431 B Save Response

Retrieve YANG data in xml

SEND_NEW_LINK

The screenshot shows the Postman interface with a REST client request configured. The URL is `https://10.10.20.48/restconf/data/ietf-interfaces:interfaces`. The method is GET. The headers tab is selected, showing 7 headers. The body tab is also selected, showing a JSON payload. The status bar indicates a 200 OK response with a 512 ms latency and 1.55 KB of data.

Headers (7)

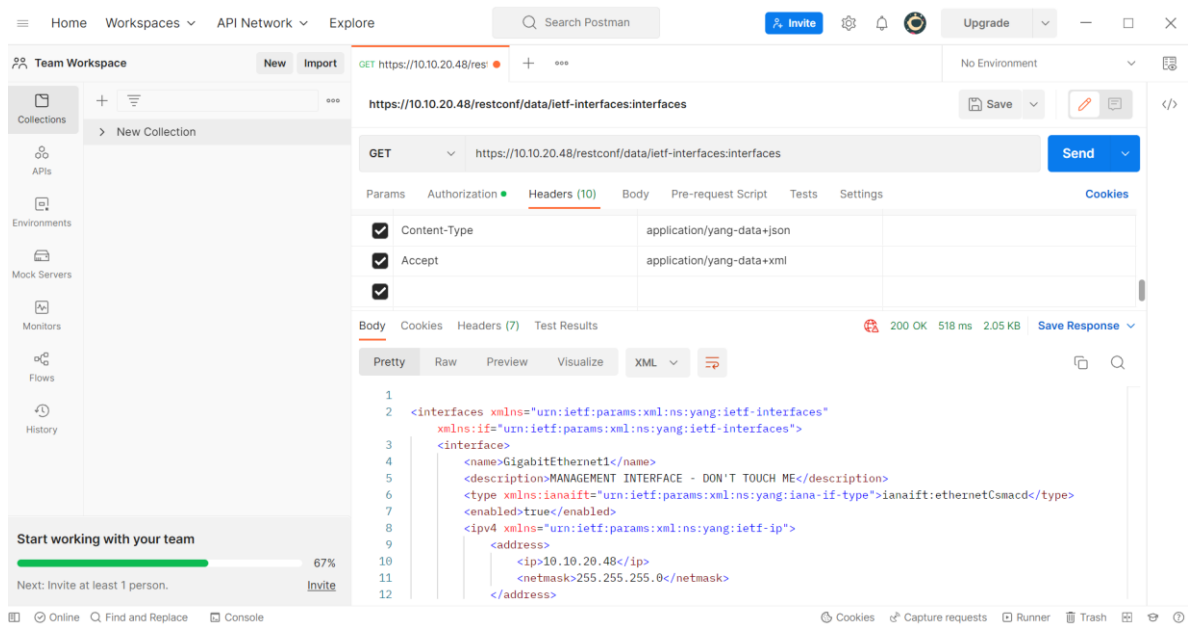
| Key | Value | Description |
|--------------|----------------------------|-------------|
| Content-Type | application/yang-data+json | |
| Accept | application/yang-data+json | |
| | | |
| | | |
| | | |
| | | |
| | | |

Body

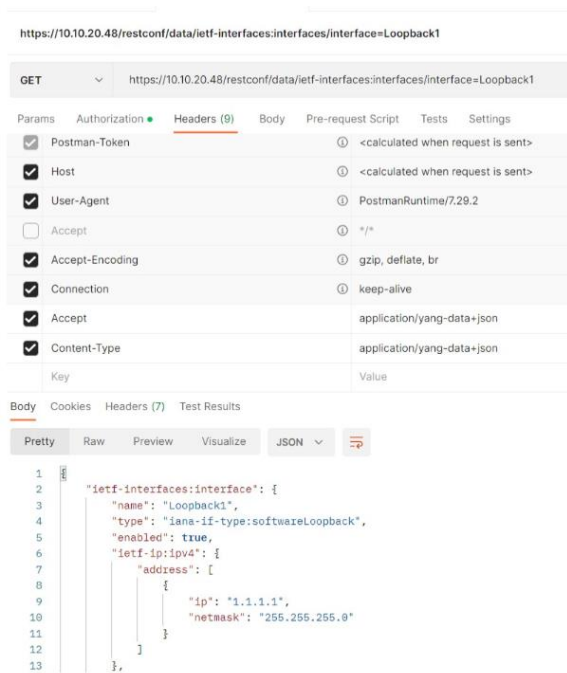
JSON

```
2 {
3   "ietf-interfaces:interfaces": {
4     "interface": [
5       {
6         "name": "GigabitEthernet1",
7         "description": "MANAGEMENT INTERFACE - DON'T TOUCH ME",
8         "type": "iana-if-type:ethernetCsmacd",
9         "enabled": true,
10        "ietf-ip:ipv4": {
11          "address": [
12            {
13              "ip": "10.10.20.48",
14              "netmask": "255.255.255.0"
15            }
16          ]
17        }
18      }
19    ]
20  }
21 }
```

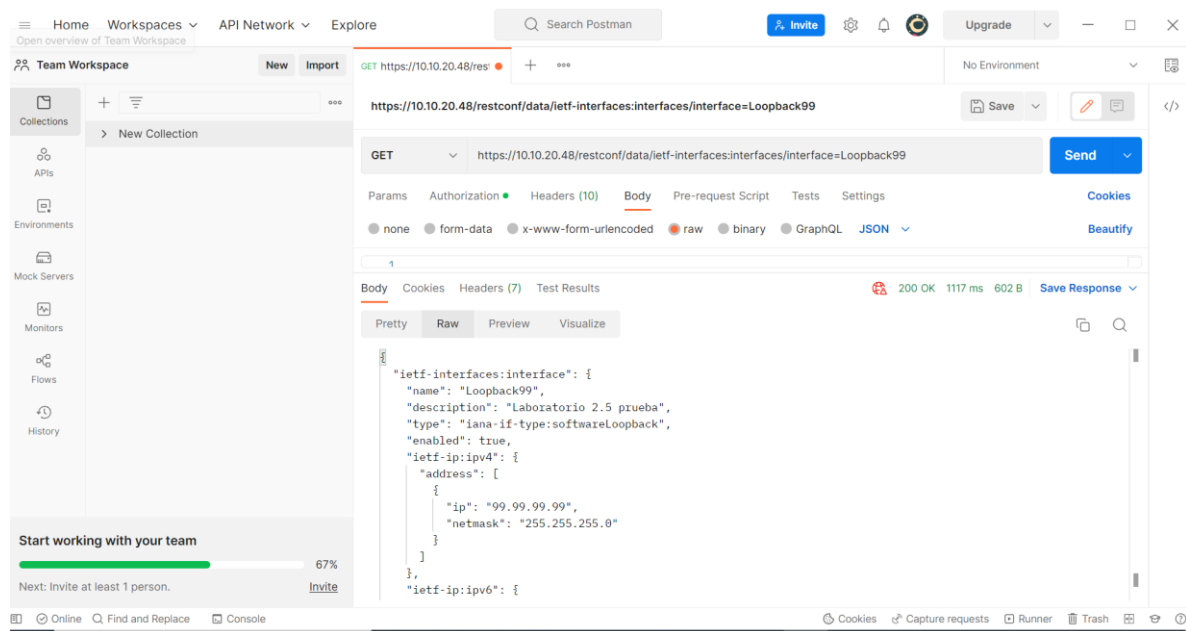
200 OK 512 ms 1.55 KB Save Response



SEND_LOOPBACK1



SEND_LOOPBACK99



Conclusión

Al realizar la practica logre comprender en cómo utilizar el potsman Como ayuda al desarrollo, permite al equipo mantener una colección actualizada de todas las llamadas de tu servicio o una colección que permita probar rápidamente la integración se logro llegar al resultado obtenido, aunque se tuvo que investigar mas a fondo sobre el poque se dice que Postman surgió originariamente como una extensión para el navegador Google Chrome. A día de hoy dispone de aplicaciones nativas para MAC y Windows y están trabajando en una aplicación nativa para Linux (disponible en versión beta).

Está compuesto por diferentes herramientas y utilidades gratuitas (en la versión free) que permiten realizar tareas diferentes dentro del mundo