Instituto tecnológico de colima

Ingeniería informática

Diego Rodolfo Diaz Morfin

Apps IOT

ANGEL VEJAR CORTES

Villa de Álvarez, colima, agosto – diciembre 2023

Introducción

En esta actividad, se exploró el manejo de solicitudes HTTP utilizando Python como lenguaje de programación para el servidor y postman como cliente HTTP. El objetivo principal fue comprender cómo interactuar con solicitudes HTTP POST y GET utilizando un servidor HTTP implementado en Python y cómo probar estas solicitudes utilizando postman.

Como tecnologías utilizadas fueron:

- Python: Se utilizó Python como lenguaje de programación para implementar un servidor HTTP.
- Postman: Es una herramienta que permite probar solicitudes HTTP de manera eficiente. Se utilizó postman para enviar solicitudes POST al servidor Python y verificar las respuestas.
- HTTP (Hypertext Transfer Protocol): Se trata del protocolo de comunicación utilizado en la World Wide Web para la transferencia de datos. Fue el protocolo principal utilizado en esta práctica para la comunicación entre el cliente (postman) y el servidor.

Desarrollo

- Descarga de la herramienta postman para probar solicitudes HTTP

Código:

- Se configura la respuesta HTTP con el código 200 (OK) y un tipo de contenido especificado que es “text/plain”.

- Se definen dos métodos que manejan las solicitudes GET y POST

```python
    Codeium: Refactor | Explain | Generate Docstring
    def do_GET(self):
        self._set_response()
        respuesta = "el valor es: " + str(contador)
        self.wfile.write(respuesta.encode())

    Codeium: Refactor | Explain | Generate Docstring
    def do_POST(self):
        content_length = int(self.headers["Content-Length"])
        post_data = self.rfile.read(content_length)

        body_json = json.loads(post_data.decode())
```

- Aquí se comprobó si el cuerpo JSON contiene las claves action y quantity las cuales se les asignaron en variables.

```python
    global contador
    if 'action' in body_json and 'quantity' in body_json:
        action = body_json['action']
        quantity = int(body_json['quantity'])

        if action == 'asc':
            contador += quantity
        elif action == 'desc':
            contador -= quantity
```

server.py M

server.py > MyHTTPRequestHandler > do_GET

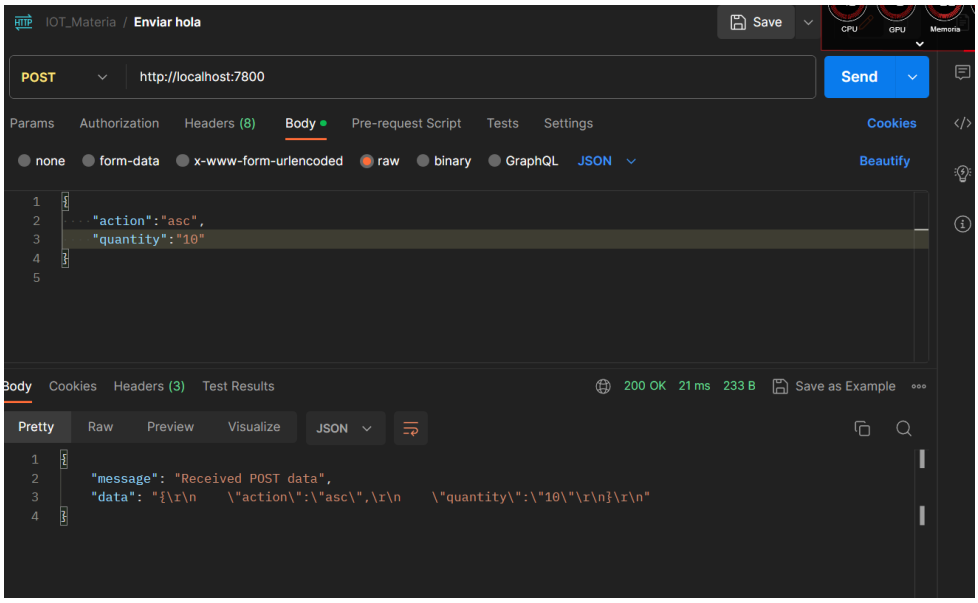
```python
        # Print the complete HTTP request
        print("\n----- Incoming POST Request -----")
        print(f"Requestline: {self.requestline}")
        print(f"Headers:\n{self.headers}")
        print(f"Body:\n{post_data.decode()}")
        print("--------------------------------")

        # Respond to the client
        response_data = json.dumps({"message": "Received POST data",
                                    "data": post_data.decode()})
        self._set_response("application/json")
        self.wfile.write(response_data.encode())

Codeium: Refactor | Explain | Generate Docstring
def run_server(server_class=HTTPServer, handler_class=MyHTTPRequestHandler, port=7800):
    server_address = ("", port)
    httpd = server_class(server_address, handler_class)
    print(f"Starting server on port {port}...")
    httpd.serve_forever()

if __name__ == "__main__":
    run_server()
```

- Server corriendo

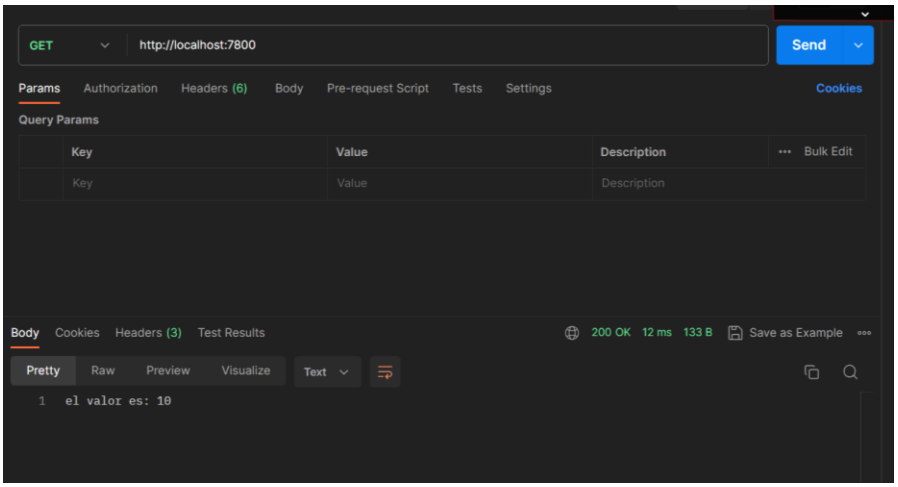
```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  COMENTARIOS                Code

[Done] exited with code=1 in 1246.793 seconds

[Running] python -u "c:\Users\rdiaz\OneDrive\Escritorio\Escuela\Sem 9\Apps_IOT\U1\server.py"
Starting server on port 7800...
```

Postman:

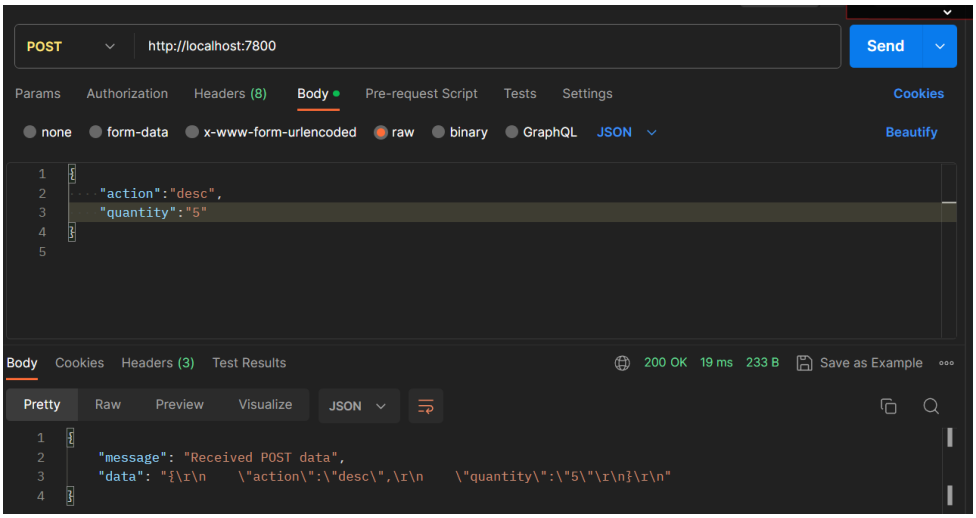
- Aquí se le manda un 10 para incrementar el valor



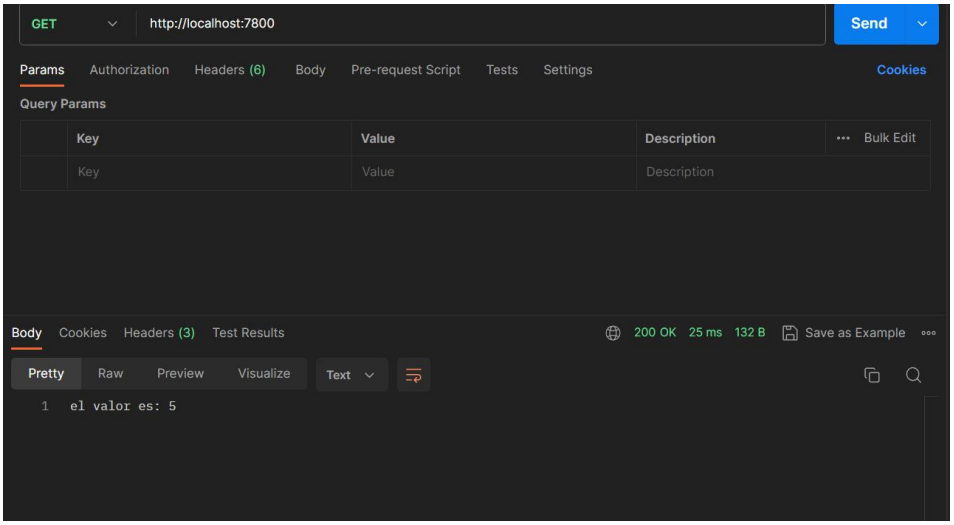
- Recibimos el valor



- Aquí se le mando un 5 para decrementar el valor



- Recibimos el valor



Conclusión

El protocolo HTTP nos ayuda en la programación web permitiendo la transferencia de datos entre clientes y servidores de manera eficiente y es esencial para la mayoría de las interacciones en línea y con ayuda de las herramientas como Python y postman facilitan el desarrollo, la depuración y las pruebas de aplicaciones basadas en HTTP. Lo cual pudimos observar en esta actividad donde comprendimos el funcionamiento de las solicitudes HTTP y como interactúan con un servidor web.