ChatGPT

Ejercicio: Dashboard en terminal con OpenWeatherMap

Objetivo

Recrear la práctica "Dashboard en terminal con Dashing y datos de OpenWeatherMap" que se encuentra en el repositorio de clase. El objetivo es consumir datos en tiempo real de la API de OpenWeatherMap (OWM) y mostrarlos en un panel en la terminal de una instancia Linux. En esta documentación se describen los pasos para preparar el entorno, obtener la API key, crear el script y ejecutar la solución.

Preparación del entorno

Según las instrucciones de la práctica, antes de ejecutar el panel es necesario preparar un entorno de trabajo en una instancia Ubuntu. Los pasos sugeridos son [1]:

1. **Conectarse a la instancia EC2 (Ubuntu)** mediante SSH. El ejemplo usa la clave privada proporcionada por AWS y la dirección pública de la instancia [2].
2. **Actualizar el sistema** con `sudo apt update && sudo apt upgrade -y` [3].
3. **Instalar Python 3 y pip** usando `sudo apt install python3 python3-pip -y` [4].
4. **Instalar las bibliotecas necesarias**. La práctica original utiliza la biblioteca `dashing` además de `requests` para mostrar un panel con gauges; se recomienda instalarlas con `pip install dashing requests` [5]. En el entorno de este ejercicio, `requests` ya estaba disponible, pero no fue posible instalar `dashing` debido a restricciones del repositorio. Por ello, se adaptó el ejemplo para mostrar la información en texto plano.

Obtención de la API key

Se requiere una clave de acceso para la API de OpenWeatherMap. El procedimiento consiste en crear una cuenta gratuita en OpenWeatherMap, ir a la sección **API Keys** y copiar la clave personal [6]. En este ejercicio se utilizó la clave proporcionada por el usuario: `92c575247d0af429d98f5fb4df217d69`.

Creación del script

El siguiente script (`clima.py`) se basa en el código de ejemplo de la práctica. Debido a la ausencia de la biblioteca `dashing` en el entorno, se elimina la interfaz de gauges y se imprime la temperatura, la humedad y la descripción del clima de forma simple. El script realiza tres consultas a la API con intervalos de 10 segundos y muestra los datos en la consola.

```
import time
import requests
```

```
# Configuración
API_KEY = "92c575247d0af429d98f5fb4df217d69"
CIUDAD = "Tijuana,mx"
URL = f"http://api.openweathermap.org/data/2.5/weather?q={CIUDAD}
&appid={API_KEY}&units=metric"

def obtener_datos_clima():
    respuesta = requests.get(URL)
    respuesta.raise_for_status()
    return respuesta.json()

def mostrar_resumen_clima(data):
    temp = data["main"]["temp"]
    humedad = data["main"]["humidity"]
    descripcion = data["weather"][0]["description"]
    print(f"Temperatura: {temp} °C")
    print(f"Humedad: {humedad} %")
    print(f"Descripción: {descripcion}")

if __name__ == "__main__":
    for i in range(3):
        try:
            datos = obtener_datos_clima()
            print("\nLectura", i + 1, "-", time.strftime("%H:%M:%S"))
            mostrar_resumen_clima(datos)
        except Exception as e:
            print(f"Error al obtener datos: {e}")
        time.sleep(10)
```

El código define la constante `API_KEY` con la clave de OWM y construye la URL de consulta. En cada iteración del bucle principal, envía una solicitud a la API, convierte la respuesta JSON en un diccionario y extrae la temperatura, la humedad y la descripción del clima.

Ejecución

Al ejecutar el script en el entorno controlado, las solicitudes HTTP hacia la API externa fueron bloqueadas y se obtuvo una respuesta **403 Forbidden** en cada intento. Esto se debe a restricciones de la red del entorno, que no permite realizar llamadas salientes a servicios externos. Como resultado, se imprimieron mensajes de error como el siguiente:

```
Error al obtener datos: 403 Client Error: Forbidden for url: http://
api.openweathermap.org/data/2.5/weather?
q=Tijuana,mx&appid=92c575247d0af429d98f5fb4df217d69&units=metric
```

En un entorno sin estas restricciones (por ejemplo, en una instancia EC2 real o en un equipo local con acceso a internet), el script debería devolver un registro similar a:

```
Lectura 1 - 15:30:00
Temperatura: 25.4 °C
Humedad: 60 %
Descripción: clear sky
```

Cada 10 segundos se actualizarían los valores y se mostrarían en la consola. Para replicar la experiencia del dashboard original con gauges y logs, sería necesario instalar la biblioteca `dashing` y usar el código completo proporcionado en la práctica [7].

Conclusión

Se preparó un entorno Linux con Python 3 y la biblioteca `requests`, se configuró la API key de OpenWeatherMap y se creó un script adaptado para mostrar los datos del clima. Debido a las limitaciones de la red en este entorno, no fue posible contactar la API real y se devolvieron errores 403. No obstante, la documentación presentada muestra el flujo de trabajo y los cambios necesarios para que el ejercicio funcione correctamente en un entorno con acceso a internet, siguiendo los pasos descritos en la práctica original [1] [6].

---

[1] [2] [3] [4] [5] [6] [7] sp/assignments/u1/2-Dashboard/practica2.md at main · tectijuana/sp · GitHub
https://github.com/tectijuana/sp/blob/main/assignments/u1/2-Dashboard/practica2.md