In [ ]:

In [1]:
```
pip install aiohttp
```

```
Requirement already satisfied: aiohttp in c:\users\anuj2\anaconda3\lib\site-packages (3.9.5)
Requirement already satisfied: aiosignal>=1.1.2 in c:\users\anuj2\anaconda3\lib\site-packages (from aiohttp) (1.2.0)
Requirement already satisfied: attrs>=17.3.0 in c:\users\anuj2\anaconda3\lib\site-packages (from aiohttp) (23.1.0)
Requirement already satisfied: frozenlist>=1.1.1 in c:\users\anuj2\anaconda3\lib\site-packages (from aiohttp) (1.4.0)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\users\anuj2\anaconda3\lib\site-packages (from aiohttp) (6.0.4)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\users\anuj2\anaconda3\lib\site-packages (from aiohttp) (1.9.3)
Requirement already satisfied: idna>=2.0 in c:\users\anuj2\anaconda3\lib\site-packages (from yarl<2.0,>=1.0->aiohttp) (3.7)
Note: you may need to restart the kernel to use updated packages.
```

In [6]:
```
import asyncio
import aiohttp
import json


API_KEY = "38f34d9bbc83a5a3ad50db45380a0183"
BASE_URL = "http://api.openweathermap.org/data/2.5/weather"


async def fetch_weather(city, unit="C"):
    url = f"{BASE_URL}?q={city}&appid={API_KEY}&units=metric"
    async with aiohttp.ClientSession() as session:
        async with session.get(url) as response:
            if response.status == 200:
                data = await response.json()
                return format_weather(data, unit=unit)
            elif response.status == 404:
                return f"City '{city}' not found."
            else:
                return f"Could not fetch weather for {city} (Status: {response.status})"


def format_weather(data, unit="C"):

    temp_celsius = data["main"]["temp"]


    if unit == "F":
        temp = temp_celsius * 9 / 5 + 32
    elif unit == "K":
        temp = temp_celsius + 273.15
    else:
        temp = temp_celsius


    city = data["name"]
    description = data["weather"][0]["description"]
    humidity = data["main"]["humidity"]


    return (f"Weather in {city}:\n"
            f"- Temperature: {temp:.2f}°{unit}\n"
            f"- Condition: {description.capitalize()}\n"
            f"- Humidity: {humidity}%\n")


async def main(cities, unit="C"):
    tasks = [fetch_weather(city, unit=unit) for city in cities]
    results = await asyncio.gather(*tasks)
    for result in results:
        print(result)


cities = ["New York", "London", "Tokyo", "Sydney"]
unit = "F"


await main(cities, unit=unit)
```

```
Weather in New York:
- Temperature: 52.77°F
- Condition: Clear sky
- Humidity: 63%

Weather in London:
- Temperature: 54.41°F
- Condition: Few clouds
- Humidity: 79%

Weather in Tokyo:
- Temperature: 63.97°F
- Condition: Few clouds
- Humidity: 75%

Weather in Sydney:
- Temperature: 64.45°F
- Condition: Overcast clouds
```

- Humidity: 83%

In [ ]: