This guide explains how to set up the required tools and dependencies for detecting network interface types (Ethernet or Wi-Fi) and fetching their IP addresses on Windows using Python. It also includes detailed instructions for installing dependencies and the necessary code.

Prerequisites

Before proceeding, ensure you have:

1. Python installed (3.8 or higher recommended).
2. Administrator privileges on your computer.
3. Visual Studio Build Tools installed.

Step 1: Install Visual Studio Build Tools (if not already installed)

Visual Studio Build Tools are required for compiling Python dependencies that interact with system-level components.

Instructions:

1. Open the **Visual Studio Installer**.

2. Select **Desktop development with C++** under the *Workloads* tab

3. In the **Installation details** section, ensure the following components are selected:

    - `MSVC v143 - VS 2022 C++ x64/x86 build tools`
    - `Windows 11 SDK`
    - `C++ CMake tools for Windows`

4. Click **Modify** or **Install** to begin the installation.

Step 2: Install Python Dependencies

After the build tools are installed, install the required Python libraries:

1. Open a command prompt or terminal.
2. Run the following commands to install `netifaces` and `wmi`:

```
pip install netifaces WMI
```

Step 3: Python Code for Network Detection

Use the following Python script to check for active network interfaces, determine whether they are Ethernet or Wi-Fi, and fetch their IP addresses:

```
import netifaces
import wmi

def check_connection_type_and_ip():
    # Initialize WMI client
    wmi_client = wmi.WMI()
    connection_type = None
    ip_address = None

    # Iterate through all network adapters with WMI to check for Wi-Fi or Ethernet
    for nic in wmi_client.Win32_NetworkAdapterConfiguration(IPEnabled=True):
        interface = nic.Description  # Description of the network interface
        ip_addresses = nic.IPAddress  # List of IP addresses associated with this interface

        # Check for Wi-Fi by looking for "Wireless" or "Wi-Fi" in the adapter name
        if "Wireless" in interface or "Wi-Fi" in interface:
            connection_type = "Wi-Fi"
        else:
            connection_type = "Ethernet"

        # Get the primary IP address if available
        ip_address = ip_addresses[0] if ip_addresses else None

        # Print interface details
        print(f"Interface: {interface}")
        print(f"Connection Type: {connection_type}")
        print(f"IP Address: {ip_address}")
        print()

    if connection_type is None:
        print("No active Ethernet or Wi-Fi connection detected.")

    return connection_type, ip_address

# Run the function
connection_type, ip_address = check_connection_type_and_ip()
```

Step 4: Running the Script

1. Save the script as `interfaces.py`.
2. Open a terminal in the directory containing the script.

3. Run the script:

   ```
   python interfaces.py
   ```

Sample Output:

When you run the script, you should see an output similar to this:

```
Interface: Intel(R) 82567LM-3 Gigabit Network Connection
Connection Type: Ethernet
IP Address: 192.168.88.225
```

Explanation of the Script

1. **WMI for Interface Information**:
   - The script uses `wmi` to list all active network adapters.
   - It checks the adapter description to determine whether it's Wi-Fi (`Wireless` or `Wi-Fi`) or Ethernet.
2. **IP Address Fetching**:
   - The IP address of each adapter is retrieved using the `IPAddress` property from the WMI network configuration.
3. **Output**:
   - The script prints the interface name, its connection type, and the associated IP address.

Troubleshooting

- If the script does not detect interfaces or gives errors, ensure:
  - You have administrator privileges.
  - The Visual Studio Build Tools installation includes the required components.
  - All Python dependencies are correctly installed.
