Brief instructions to run provided client applications

To run the Python MQTT client applications, the following steps should be followed:

Prerequisites

1. **Python Installation**: Ensure Python 3.6 or higher is installed. The user can verify the version on their machine before proceeding.
2. **Required Library**: These programs require the `paho-mqtt` library, which facilitates MQTT communication in Python. This library may need to be installed, as it is not part of Python's standard libraries.

Preparing Certificate and Key Files

1. **Certificate Placement**: The user should have three certificate files:
    - **Root CA Certificate** named `root.crt`
    - **Client Certificate** named `IntCert.crt`
    - **Client Key** named `intcert.key`

    These files should either be placed in the same directory as the Python scripts or, if stored elsewhere, the file paths in the scripts should be updated accordingly.

2. **Broker Configuration**: The user must ensure that the MQTT broker, configured to use the certificate-based authentication, is active and accessible on the specified IP and port (136.186.230.70 on port 8883).

Running the Scripts

1. **Publisher Script**: This script will generate and publish temperature data to the specified MQTT topics. After starting the script, it should display messages confirming the successful publication of data.
2. **Subscriber Script**: Running this script in a separate terminal will allow it to connect to the broker and start listening for incoming messages. Messages received from the publisher should appear in the subscriber's output.

Authentication

The scripts are configured to connect using the MQTT broker's username "abid" and password "password." The user should verify these credentials with the broker or update them if necessary.

Expected Output

Both the publisher and subscriber scripts will output connection confirmations, and the publisher will display any messages sent to the topics, while the subscriber will show any received messages. This setup should enable seamless communication between the publisher and subscriber clients over the secured MQTT connection.