4.2. Лабораторная работа: MQTT-клиент в Python

Site: Samsung Innovation Campus
Course: Internet of Things
Book: 4.2. Лабораторная работа: MQTT-клиент в Python

Printed by: Антон Файтельсон
Date: Saturday, 21 October 2023, 7:38 PM

Table of contents

4.2.1. Пример взаимодействия с сервером на Python

До нынешнего момента работа с MQTT-сервером производилась просто в консоли при помощи команд mosquitto_sub и mosquitto_pub — это удобно для одноразовых задач вроде считывания данных или отправки тестовых команд.

Для взаимодействия с MQTT-сервером существуют различные библиотеки. Мы рассмотрим библиотеки из проекта Eclipse Paho. Под крылом проекта Paho разрабатываются открытые реализации протокола MQTT в виде библиотек для почти всех популярных языков программирования: существуют реализации для C, C++, Java, Go, C#, Python, Javascript.

Рассмотрим пример работы с Paho в Python. Саму библиотеку очень просто установить:

```
pip install paho-mqtt
```

Попробуйте запустить тестовую программу. Она взята из примера в репозитории без изменений. Её функционал очень прост: вывод на экран сообщений из определённого топика.

```
import paho.mqtt.client as mqtt

# The callback for when the client receives a CONNACK response from the server.
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    # Subscribing in on_connect() means that if we lose the connection and
    # reconnect then subscriptions will be renewed.
    client.subscribe("$SYS/#")

# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect("iot.eclipse.org", 1883, 60)

# Blocking call that processes network traffic, dispatches callbacks and
# handles reconnecting.
# Other loop*() functions are available that give a threaded interface and a
# manual interface.

client.loop_forever()
```

Изучите текст примера и выделите для себя ключевые моменты:

- Где и в какой момент происходит подписка на топик?
- Где происходит подключение к серверу?
- Где обрабатывается входящее сообщение?

Модифицируйте программу так, чтобы она работала с вашим локальным сервером и подписывалась на тот топик, который вас интересует.

4.2.2. Отправка сообщений в Python

Из документации библиотеки Paho можно узнать, что функция для отправки сообщения выглядит так:

```
publish(topic, payload=None, qos=0, retain=False)
```

Все параметры понятны, кроме последнего:

- topic — топик, в который идет сообщение
- payload — текст сообщения
- qos — качество обслуживания (Quality of Service)
- retain — новый параметр. Если он имеет значение «Истина», то сообщение будет «храниться до востребования», если клиент сейчас отключен, то после включения он все равно это сообщение получит, поскольку оно будет храниться на сервере. В нашем случае информация может устареть, поэтому разумно установить его в false. Данный параметр хорош для передачи тех сообщений, которые не устаревают: например, ценные данные мониторинга.

Задание:

Добавьте в вашу программу на Python публикацию сообщений в некий топик (какой - придумайте сами).

К примеру, можно сделать программный генератор сообщений - раз в несколько секунд отправление данных о влажности и температуре, подобно тому, как это делает реальное устройство. Бесконечный цикл loop_forever() в начале программы можно заменить на loop_start() и loop_stop() (об этом написано в документации).

4.2.3. Топики в MQTT - задание на самостоятельное выполнение

Используя локальный MQTT-сервер, выполните следующие задания:

1. Проверьте, чувствительны ли названия топиков к регистру.
2. Можно ли послать сообщение сразу всем субтопикам определённого топика? Предположим, что у вас есть топик light, в нём топики lamp1 и lamp2, и вы хотите послать сигнал всем лампочкам включиться.
3. Будет ли работать подписка на топик, если в конце названия топика добавить символ «/»? А если в начале?
4. Что будет, если вызвать mosquitto_pub с ключом -l?
5. Изучите, что такое retain. Отправьте сообщение с ключом -r, но важно, что в этот момент ещё не должен быть запущен подписчик. Затем запустите подписку. Что произойдет? Что будет, если подписчик закрыт и затем запущен заново? Что, если оставить второе сообщение с тем же ключом? Чтобы вернуть всё к прежнему состоянию, отправьте пустое сообщение с ключом -r.
6. Изучите, что такое «Последняя воля и завещание» (Last Will and Testament). Для этого при публикации не задавайте тело сообщения, но укажите параметры --will-topic и --will-payload - название топика и собственно текст «завещания». В результате, если вы в другом окне оформите подписку на этот топик, и издатель (mosquitto_pub) незапланированно отключится, то подписчик (mosquitto_sub) получит от сервера «завещание». Обратите внимание, что команда mosquitto_pub в чистом виде отправляет сообщение и сразу отключается без аварийного завершения, поэтому у вас не получится сделать это задание сразу. Подумайте, как смоделировать случай, когда издатель подключается на какое-то время и затем внезапно отключается.