Lab #4

IoT

Course : CPE 371 - Big Data Engineering
Instructor : Rajchawit Sarochawikasit
Assignment : Lab Assignment 4: IoT

Group Members

65070503444	Jutamas Kaewcheunchai
65070503448	Thanakit Chokbunsuwan
65070503470	Arita Tragulmalee
65070503480	Yuil Tripathee

[please find the lab instruction and `.ipynb` files in LEB2, Here's a short brainstorming]

Problem Statement

Modify the subscriber file for writing the received data from the publisher into a CSV file.

Publisher code

These codes are extracted from the Interactive Ipython notebook:

Code:

```
import paho.mqtt.client as mqtt
MQTT_HOST = "broker.mqttdashboard.com"
MQTT_PORT = 1883
#MQTT_HOST = "mqtt-dashboard.com"
#MQTT_PORT = 8884
MQTT_KEEPALIVE_INTERVAL = 60 #60 seconds
MQTT_TOPIC = "CPE_DEMO_HOUSE/bdt"

def on_publish(self,client,userdata):
    print("Message Published...")


client = mqtt.Client()
client.on_publish = on_publish
client.connect(MQTT_HOST,MQTT_PORT,MQTT_KEEPALIVE_INTERVAL)
```

Output:

```
0
```

Code:

```
MQTT_MSG = "Hello World!"
client.publish(MQTT_TOPIC,MQTT_MSG,qos=0,retain=True)
```

Output:

```
Message Published...
<paho.mqtt.client.MQTTMessageInfo at 0x7001030f5170>
```

Code to publish as JSON format:

```
import json
message = { "Temperature":35,
            "Humidity":60,
            "Light":146}

MQTT_MSG = json.dumps(message)
client.publish(MQTT_TOPIC,MQTT_MSG,qos=0,retain=True)
```

Output:

```
Message Published...
<paho.mqtt.client.MQTTMessageInfo at 0x7001030f6c50>
```

This block can be run multiple times to publish several messages.

Subscriber code

Here's the modified code that prints the message and dumps the message to append on a CSV file.

```
import paho.mqtt.client as mqtt
import csv
import json
from datetime import datetime
import portalocker

MQTT_HOST = "broker.mqttdashboard.com"
MQTT_PORT = 1883
MQTT_KEEPALIVE_INTERVAL = 45 #45 seconds
MQTT_TOPIC = "CPE_DEMO_HOUSE/#"
CSV_FILE = "sensor_data.csv"

def on_connect(self,client, userdata, rc):
    print("MQTT Connected.")
    self.subscribe(MQTT_TOPIC)
    # add more subscribe ...
    # e.g.;
```

```
    # self.subscribe("CPE_DEMO_HOUSE/room2/#")

def on_message(client, userdata, msg):
    print("message received...")
                print(msg.topic     +     "     "+     str(msg.qos)     +     "     "     +
str(msg.payload.decode("utf-8","strict")))

     # Try parsing the message payload as JSON
    try:
        data = json.loads(msg.payload.decode("utf-8"))
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        topic = msg.topic
        temperature = data.get("Temperature", "N/A")
        humidity = data.get("Humidity", "N/A")
        light = data.get("Light", "N/A")

        # Open CSV in append mode and write the data
        with open(CSV_FILE, mode='a', newline='') as file:
            # Acquire file lock
            portalocker.lock(file, portalocker.LOCK_EX)

            writer = csv.writer(file)
            # Write headers if the file is empty
            if file.tell() == 0:
                        writer.writerow(["Timestamp", "Topic", "Temperature", "Humidity",
"Light"])
            # Write the data row
            writer.writerow([timestamp, topic, temperature, humidity, light])
            # Release file lock
            portalocker.unlock(file)

              print(f"Data  written  to  {CSV_FILE}:  {timestamp},  {topic},  {temperature},
{humidity}, {light}")

    except json.JSONDecodeError as e:
        print(f"Failed to decode JSON: {e}")

client = mqtt.Client()

client.on_connect = on_connect
client.on_message = on_message

client.connect(MQTT_HOST)
client.loop_forever()
```

Output:

```
MQTT Connected.
message received...
```

```
CPE_DEMO_HOUSE/ 0 {"Temperature": 32, "Humidity": 54, "Light": 102}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/, 32, 54, 102
message received...
CPE_DEMO_HOUSE/Hill 0 {"Temperature": 20, "Humidity": 30, "Light": 200}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/Hill, 20, 30, 200
message received...
CPE_DEMO_HOUSE/room 0 {"Temperature": 32, "Humidity": 54, "Light": 102}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/room, 32, 54, 102
message received...
CPE_DEMO_HOUSE/12 0 {"Temperature": 32, "Humidity": 54, "Light": 102}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/12, 32, 54, 102
message received...
CPE_DEMO_HOUSE/BACKROOMS 0 OUR GROUP IS
Failed to decode JSON: Expecting value: line 1 column 1 (char 0)
message received...
CPE_DEMO_HOUSE/room2 0 {"Temperature": 20, "Humidity": 30, "Light": 200}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/room2, 20, 30, 200
message received...
CPE_DEMO_HOUSE/clowns 0 {"Temperature": 69, "Humidity": 96, "Light": 39}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/clowns, 69, 96, 39
message received...
CPE_DEMO_HOUSE/ARM 0 your
Failed to decode JSON: Expecting value: line 1 column 1 (char 0)
message received...
CPE_DEMO_HOUSE/room1 0 {"Temperature": 35, "Humidity": 60, "Light": 138}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/room1, 35, 60, 138
message received...
CPE_DEMO_HOUSE/r 0 {"Temperature": 32, "Humidity": 54, "Light": 102}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/r, 32, 54, 102
message received...
CPE_DEMO_HOUSE/room4 0 ARM
Failed to decode JSON: Expecting value: line 1 column 1 (char 0)
message received...
CPE_DEMO_HOUSE/FOREST 0 HELP ME
Failed to decode JSON: Expecting value: line 1 column 1 (char 0)
message received...
CPE_DEMO_HOUSE/room3 0 OUR GROUP IS
Failed to decode JSON: Expecting value: line 1 column 1 (char 0)
message received...
CPE_DEMO_HOUSE/room17 0 {"Temperature": 35, "Humidity": 60, "Light": 138}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/room17, 35, 60, 138
message received...
CPE_DEMO_HOUSE/room19 0 {"Temperature": 32, "Humidity": 54, "Light": 102}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/room19, 32, 54, 102
message received...
CPE_DEMO_HOUSE/Tong 0 foolish
Failed to decode JSON: Expecting value: line 1 column 1 (char 0)
message received...
CPE_DEMO_HOUSE/thunder 0 Hello World!
```

```
Failed to decode JSON: Expecting value: line 1 column 1 (char 0)
message received...
CPE_DEMO_HOUSE/room100 0 {"Temperature": 35, "Humidity": 60, "Light": 138}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/room100, 35, 60, 138
message received...
CPE_DEMO_HOUSE/NUT 0 {"Temperature": 20, "Humidity": 30, "Light": 200}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/NUT, 20, 30, 200
message received...
CPE_DEMO_HOUSE/TEST 0 Muahahhaha
Failed to decode JSON: Expecting value: line 1 column 1 (char 0)
message received...
CPE_DEMO_HOUSE/bdt 0 {"Temperature": 35, "Humidity": 60, "Light": 146}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/bdt, 35, 60, 146
message received...
CPE_DEMO_HOUSE/roo 0 {"Temperature": 32, "Humidity": 54, "Light": 102}
Data written to sensor_data.csv: 2024-09-24 11:28:22, CPE_DEMO_HOUSE/roo, 32, 54, 102
```

Exception handling filters only those data that can be appended to a CSV file.

Here's the snapshot for the CSV file:

Delimiter: ,

	Timestamp	Topic	Temperature	Humidity	Light
1	2024-09-24 11:27:59	CPE_DEMO_HOUSE/bdt	35	60	146
2	2024-09-24 11:28:02	CPE_DEMO_HOUSE/bdt	35	60	146
3	2024-09-24 11:28:22	CPE_DEMO_HOUSE/	32	54	102
4	2024-09-24 11:28:22	CPE_DEMO_HOUSE/Hill	20	30	200
5	2024-09-24 11:28:22	CPE_DEMO_HOUSE/room	32	54	102
6	2024-09-24 11:28:22	CPE_DEMO_HOUSE/12	32	54	102
7	2024-09-24 11:28:22	CPE_DEMO_HOUSE/room2	20	30	200
8	2024-09-24 11:28:22	PE_DEMO_HOUSE/clowns	69	96	39
9	2024-09-24 11:28:22	CPE_DEMO_HOUSE/room1	35	60	138
10	2024-09-24 11:28:22	CPE_DEMO_HOUSE/r	32	54	102
11	2024-09-24 11:28:22	PE_DEMO_HOUSE/room17	35	60	138
12	2024-09-24 11:28:22	PE_DEMO_HOUSE/room19	32	54	102
13	2024-09-24 11:28:22	E_DEMO_HOUSE/room100	35	60	138
14	2024-09-24 11:28:22	CPE_DEMO_HOUSE/NUT	20	30	200
15	2024-09-24 11:28:22	CPE_DEMO_HOUSE/bdt	35	60	146
16	2024-09-24 11:28:22	CPE_DEMO_HOUSE/roo	32	54	102