

Họ Tên: Nguyễn Thị Minh Hoa

MSSV: K205480106054

Đề Bài: Random nhiệt độ, độ ẩm

Yêu cầu

1. Trình bày thuật toán xử lý của api, ý nghĩa
2. Mô tả các bước cài đặt+ snap màn hình minh họa.
3. Mô tả quá trình chạy demo, hiểu được luồng xử lý dữ liệu. Hình ảnh minh họa
4. Kết luận: đã tìm hiểu được những kỹ thuật gì? Đã cài đặt và cấu hình thành công phần mềm nào? Đã tạo đc api gì? Đã phối hợp các kỹ thuật lập trình gì để đạt được điều gì? Kết quả cuối cùng xấu đẹp ra sao?

Những mục chính đã làm

- Tạo file python sử dụng fash api tạo dữ liệu random cho nhiệt độ và độ ẩm
- Sử dụng node-red lấy dữ liệu từ địa chỉ local của fash api
- Sử dụng asp dot net để lấy dữ liệu và vẽ biểu đồ

Quá trình làm bài

1. Cài đặt visua code, Fastapi, node-red, sql
 - Cài đặt fastapi: mở terminal thực hiện câu lệnh
pip install fastapi
pip install uvicorn
 - Cài đặt node-red: cần tải nodejs trước sau đó mới cài đặt được node-red bằng câu lệnh
npm install -g --unsafe-perm node-red
2. Kết nối node-red với sql
Cài đặt **node-red-contrib-mssql-plus** sau đó kết nối với sql

Name	demo
Server	127.0.0.1
Port	1433
Username	sa
Password	*****
Domain	
Database	demo
TDS Version	7_4 (SQL Server 2012 ~ 2022) ▼
Use Encryption?	<input checked="" type="checkbox"/>
SQL Databases hosted on Azure will need this checked.	
Trust Certificate?	<input checked="" type="checkbox"/>

3. Viết code python fastapi để tạo dữ liệu random

Đầu tiên tạo dữ liệu random cho nhiệt độ, độ ẩm

```
from fastapi import FastAPI
import random

app = FastAPI()

# Hàm để sinh ngẫu nhiên giá trị nhiệt độ
def generate_random_temperature():
    return round(random.uniform(15, 30), 2)

# Hàm để sinh ngẫu nhiên giá trị độ ẩm
def generate_random_humidity():
    return round(random.uniform(60, 95), 2)

@app.get("/")
async def read_data():
    return "Hello"

@app.get("/data")
async def read_data():
    # Gán giá trị nhiệt độ và độ ẩm bằng hàm sinh ngẫu nhiên
    temperature = generate_random_temperature()
    humidity = generate_random_humidity()
    return {"temperature": temperature, "humidity": humidity}
```

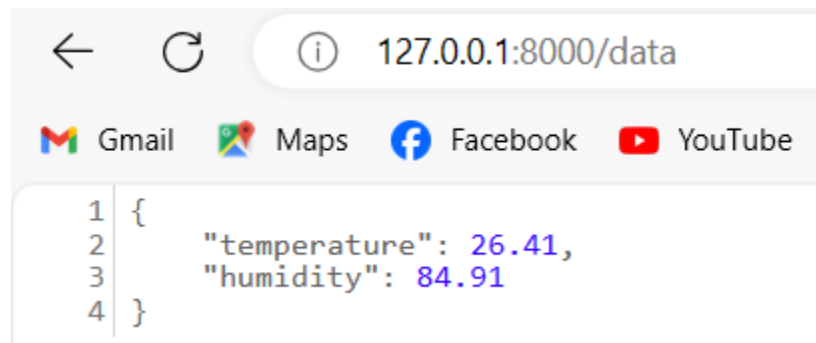
4. Chạy file main

Sau khi chạy sẽ trả về 1 chuỗi dạng json

```

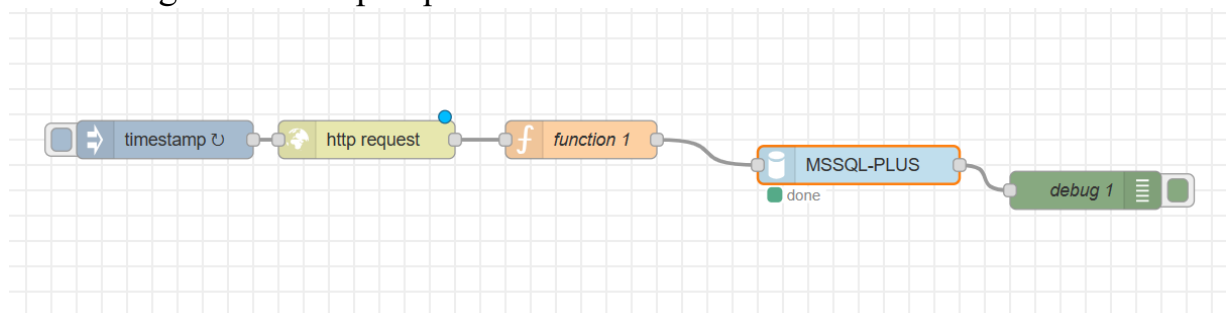
PS D:\python\pyDDC> uvicorn main:app
INFO:      Started server process [12404]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
INFO:      Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:      127.0.0.1:49791 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:49791 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:49791 - "GET / HTTP/1.1" 200 OK
INFO:      127.0.0.1:49792 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49799 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49793 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49870 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49874 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49897 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49905 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49909 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49919 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49935 - "GET /data HTTP/1.1" 200 OK
INFO:      127.0.0.1:49940 - "GET /data HTTP/1.1" 200 OK

```



5. Dùng node-red lấy dữ liệu

Dán đường link vào http request



⚙️ **Properties**

Method GET

🌐 URL http://localhost:8000/data

Set thời gian lấy dữ liệu và lưu vào database trong timestamp

+ add

☒ Inject once after 0.1 seconds, then

🔄 Repeat interval

every 30 seconds

6. Sau đó viết funcion để lưu dữ liệu vào database

```
msg.payload = `insert into history(sid, value)
values (1,${msg.payload.temperature}); insert
into history(sid, value) values
(2,${msg.payload.humidity})`
return msg;
```

7. Dữ liệu đã được lưu vào database

	id	sid	value	time
16	20	2	93	2024-05-14 22:42:35.950
17	21	1	25	2024-05-14 22:43:06.100
18	22	2	85	2024-05-14 22:43:06.103
19	23	1	26	2024-05-14 22:43:36.730
20	24	2	90	2024-05-14 22:43:36.737
21	25	1	30	2024-05-14 22:44:06.110
22	26	2	88	2024-05-14 22:44:06.110
23	27	1	26	2024-05-14 22:44:36.063
24	28	2	88	2024-05-14 22:44:36.067
25	29	1	27	2024-05-14 22:45:06.043
26	30	2	80	2024-05-14 22:45:06.043
27	31	1	28	2024-05-14 22:45:36.053
28	32	2	88	2024-05-14 22:45:36.053
29	33	1	28	2024-05-14 22:46:06.110
30	34	2	82	2024-05-14 22:46:06.113
31	35	1	26	2024-05-14 22:46:36.090
32	36	2	85	2024-05-14 22:46:36.117
33	37	1	27	2024-05-14 22:47:06.090
34	38	2	81	2024-05-14 22:47:06.090
35	39	1	27	2024-05-14 22:47:36.117
36	40	2	85	2024-05-14 22:47:36.120
37	41	1	29	2024-05-14 22:48:06.140
38	42	2	87	2024-05-14 22:48:06.140
39	43	1	28	2024-05-14 22:48:36.150
40	44	2	87	2024-05-14 22:48:36.150
41	45	1	30	2024-05-14 22:49:06.147
42	46	2	89	2024-05-14 22:49:06.150

8. Viết SP để trả về dữ liệu dạng json

```
ALTER PROCEDURE [dbo].[SP_thoi_tiet]
    @action varchar(10)
AS
BEGIN
    IF (@action='get_data')
    BEGIN
        DECLARE @json NVARCHAR(MAX) = '';

        -- Xây dựng chuỗi JSON
        SELECT @json += FORMATMESSAGE(N'{id':%d,"sid":%d,"value":%d},' , id, [sid],[value])
        FROM history;

        -- Xóa dấu phẩy thừa ở cuối chuỗi nếu có
        IF LEN(@json) > 0
        BEGIN
            SET @json = LEFT(@json, LEN(@json) - 1);
        END

        -- Bao chuỗi bằng dấu ngoặc vuông để tạo thành mảng JSON
        SET @json = '[' + @json + ']';

        -- Hiển thị kết quả
        SELECT @json;
    END
END
```

9. Sử dụng asp dot net để lấy dữ liệu vẽ biểu đồ

Em đã sử dụng bài của thầy đã gửi trong nhóm và chỉnh sửa 1 số phần như file api.aspx.cs, html, css, js để đẩy dữ liệu lên web và vẽ biểu đồ

```
namespace demo_api_57kmt
{
    public partial class api : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            //lấy biến action gửi lên
            string action = Request["action"];

            //tùy theo action mà gọi hàm tương ứng để xử lý
            switch (action)
            {
                case "get_data":
                    get_data();
                    break;
            }
        }

        // Chuỗi kết nối tới cơ sở dữ liệu
        string connectionString = "Data Source=127.0.0.1,1433;Initial Catalog=demo;User Id=sa;Password=1234;";

        protected void get_data()
        {
            // Tạo kết nối đến cơ sở dữ liệu
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                // Tạo đối tượng SqlCommand để thực thi stored procedure
                using (SqlCommand command = new SqlCommand("SP_thoi_tiet", connection))
                {
                    // Thiết lập CommandType là StoredProcedure
                    command.CommandType = CommandType.StoredProcedure;
                    command.Parameters.Add("@action", SqlDbType.VarChar).Value = "get_data";

                    try
```

Tạo chuỗi kết nối với database, tạo đối tượng sql command để gọi store procedure và thực thi store procedure trả về một chuỗi json