

FASTAPI-NODERED-MSSQL

Họ Tên: Hà Trung Hiếu K205480106045

1 Bài tập về nhà cần làm trên nhiều nền tảng:

mssql: tạo trước db, tables, sp_ để lưu dữ liệu , và để web gọi data dùng vẽ biểu đồ

python+fastAPI => tạo ra 1 api. api này chạy qua uvicorn trên console

nodered : gọi được api ở bước trên, trong nodered mới là nơi để lưu data vào db

viết web: để lấy dữ liệu từ db ra vẽ biểu đồ

2. Sau khi cài các thư viện cần thiết thì em tạo ra file python để lấy api thời tiết và chạy trên uvicorn.

```
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
main.py 3 x
A: > main.py > ...
1 from fastapi import FastAPI, HTTPException, Query
2 import requests
3 from urllib.parse import urlencode
4 import json
5
6 app = FastAPI()
7
8 def call_api(city: str, api_key: str):
9     try:
10         city = unicode(city)
11         url = f'https://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}'
12         response = requests.get(url)
13         if response.status_code == 200:
14             return response.json()
15         else:
16             raise HTTPException(status_code=response.status_code, detail=response.text)
17     except Exception as e:
18         raise HTTPException(status_code=500, detail=str(e))
19
20 @app.get("/weather/")
21 async def get_weather(city: str = Query(...), api_key: str = "e19e78b40afb3f90ff01b81d342fd1a5"):
22     try:
23         data = call_api(city, api_key)
24         humidity = data['main']['humidity']
25         pressure = data['main']['pressure']
26         wind = data['wind']['speed']
27         description = data['weather'][0]['description']
28         temp = data['main']['temp']
29
30         return {
31             "city": city,
32             "temperature": temp,
33             "pressure": pressure,
34             "humidity": humidity,
35             "wind_speed": wind,
36             "description": description
37         }
38     except Exception as e:
39         raise HTTPException(status_code=500, detail=str(e))
40
41 if __name__ == '__main__':
42     uvicorn.run(app, host='0.0.0.0', port=8000)
```

```
C:\Windows\System32\cmd.e x + v
File "C:\Users\hatrg\AppData\Local\Programs\Python\Python312\Lib\asyncio\runners.py",
n
return runner.run(main)
File "C:\Users\hatrg\AppData\Local\Programs\Python\Python312\Lib\asyncio\runners.py",
n
raise KeyboardInterrupt()
KeyboardInterrupt
INFO: Stopping reloader process [5656]
A:\>uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['A:\\']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [13276] using WatchFiles
INFO: Started server process [7364]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:50722 - "GET / HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:50722 - "GET /weather/?city=Bac%20Giang HTTP/1.1" 200 OK
```

127.0.0.1:8000/weather/?city=Bac%20Giang

Tạo bản in đẹp

```
{ "city": "Bac Giang", "temperature": 301.19, "pressure": 1012, "humidity": 78, "wind_speed": 3.64, "description": "light rain" }
```

3. Tạo cơ sở dữ liệu, tạo bảng và stored procedure

| | Column Name | Data Type | Allow Nulls |
|---|-------------|-----------|-------------------------------------|
| ▶ | city | nchar(10) | <input checked="" type="checkbox"/> |
| | temperature | nchar(10) | <input checked="" type="checkbox"/> |
| | pressure | nchar(10) | <input checked="" type="checkbox"/> |
| | humidity | nchar(10) | <input checked="" type="checkbox"/> |
| | wind_speed | nchar(10) | <input checked="" type="checkbox"/> |
| | description | nchar(10) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

```

CREATE PROCEDURE SP_weather
    @city NVARCHAR(50),
    @temperature DECIMAL(18,2),
    @pressure INT,
    @humidity INT,
    @wind_speed DECIMAL(18,2),
    @description NVARCHAR(100)
AS
BEGIN
    DECLARE @json NVARCHAR(MAX);

    SET @json = N'{"city":"' + @city + '", "temperature":"' + CONVERT(NVARCHAR(50), @temperature) + '", "pressure":"' + CONVERT(NVARCHAR(50), @pressure)
    + '", "humidity":"' + CONVERT(NVARCHAR(50), @humidity) + '", "wind_speed":"' + CONVERT(NVARCHAR(50), @wind_speed) + '", "description":"' + @description + '"}';

    SELECT @json AS json;
END;

```

3. Sử dụng node red để lưu vào mssql

Edit http request node

Delete

Cancel

Done

⚙ Properties



☰ Method GET ▼

🌐 URL `http://127.0.0.1:8000/weather/?city={{payload.city}}`

Payload Ignore ▼

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

☐ Only send non-2xx responses to Catch node

☐ Disable strict HTTP parsing

← Return a UTF-8 string ▼

☰ Headers

Edit function node

Delete

Cancel

Done

Properties

Name

function 6

Setup

On Start

On Message

On Stop

```
10
11 // Trích xuất các giá trị từ dữ liệu JSON
12 var city = msg.payload.city;
13 var temperature = msg.payload.temperature;
14 var pressure = msg.payload.pressure;
15 var humidity = msg.payload.humidity;
16 var wind_speed = msg.payload.wind_speed;
17 var description = msg.payload.description;
18
19 // Chuẩn bị dữ liệu cho việc hiển thị
20 msg.payload = {
21   city: city,
22   temperature: temperature,
23   pressure: pressure,
24   humidity: humidity,
25   wind_speed: wind_speed,
26   description: description
27 };
28
29 return msg;
30
```

Edit MSSQL node

Delete

Cancel

Done

Properties

Connection

hi

Name

Name

Query mode

Execute Procedure

Query

Editor

1 SP_weather

Parameters

Editor



Input

Name wind

Type Decin

msg. payload.wind.speed



Input

Name humidity

Type int

msg. payload.main.humid



+ add

Parse Mustache



Output property

msg. payload

Output type

Original output

Error Handling

Throw error