

BÀI TẬP PYTHON

14/5/2024

HỌ TÊN: NGUYỄN THỊ THẢO

LỚP: K56KMT.01

Yêu cầu bài toán:

- 1: gen scripts sql : struct + data ;
- 2. code py fastAPI;
- 3: flow.json exported của node-red
- 4. project web (asp dot net) : html + js + css + aspx + aspx.cs (api)

1. Gen scripts sql: struct + data.

```
SQLQuery3.sql - TH....THAYCOP (sa (53))  SQLQuery2.sql - TH....THAYCOP (sa (52))  THAO123.THAYCOP - dbo.weather

USE [THAYCOP]
GO

/***** Object: Table [dbo].[weather]    Script Date: 5/14/2024 9:04:36 PM *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[weather](
    [city] [nchar](255) NULL,
    [date] [date] NULL,
    [temp] [float] NULL
) ON [PRIMARY]
GO
```

SQLQuery2.sql - TH....THAYCOP (sa (52)) THAO123.THAYCOP - dbo.weather

```
SELECT TOP (1000) [city]
               , [date]
               , [temp]
FROM [THAYCOP].[dbo].[weather]
```

100 %

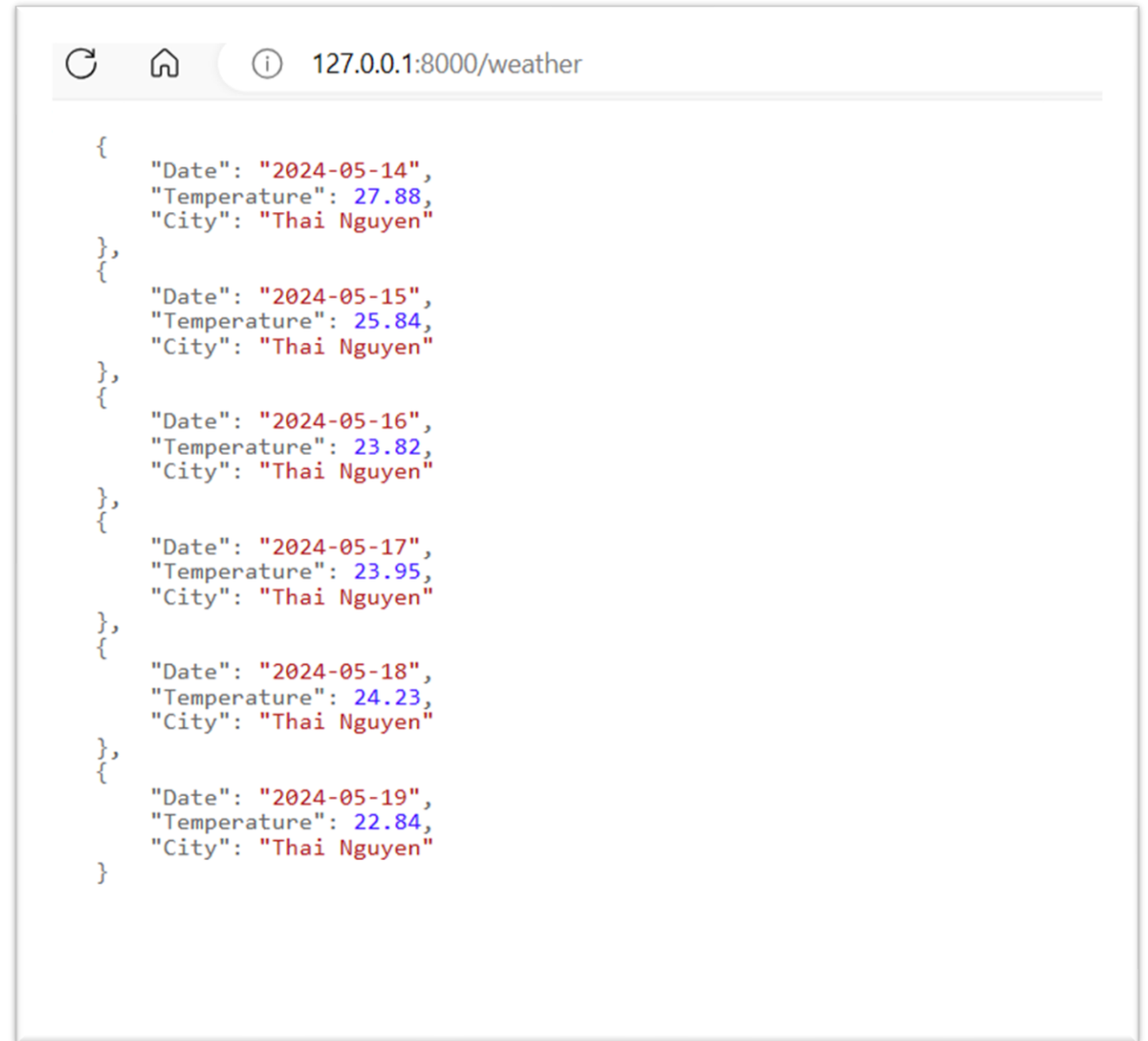
Results Messages

	city	date	temp
1	Thai Nguyen	2024-05-14	30.88
2	Thai Nguyen	2024-05-15	25.75
3	Thai Nguyen	2024-05-16	23.61
4	Thai Nguyen	2024-05-17	23.78
5	Thai Nguyen	2024-05-18	24.59
6	Thai Nguyen	2024-05-19	23.49

2. Tạo file.py sử dụng fastapi có hàm nhiệt độ

```
save_data.py ×  
1 import requests  
2 import pyodbc  
3 import logging  
4  
5 # Cấu hình logging  
6 logging.basicConfig(level=logging.INFO)  
7 logger = logging.getLogger(__name__)  
8  
9 def fetch_weather_data(url):  
10     try:  
11         response = requests.get(url)  
12         response.raise_for_status()  
13         data = response.json()  
14         logger.info("Successfully fetched data from the URL.")  
15         return data  
16     except requests.exceptions.RequestException as e:  
17         logger.error(f"Error fetching data from {url}: {e}")  
18         return None  
19  
1 usage new *
```

Sau khi chạy file sẽ trả về một chuỗi dạng json trên local của máy



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/weather'. The main content area shows a JSON array of five weather objects for the city of Thai Nguyen, spanning from May 14th to May 19th, 2024. The JSON is formatted with syntax highlighting: strings are in red, numbers are in blue, and punctuation is in black.

```
{
  "Date": "2024-05-14",
  "Temperature": 27.88,
  "City": "Thai Nguyen"
},
{
  "Date": "2024-05-15",
  "Temperature": 25.84,
  "City": "Thai Nguyen"
},
{
  "Date": "2024-05-16",
  "Temperature": 23.82,
  "City": "Thai Nguyen"
},
{
  "Date": "2024-05-17",
  "Temperature": 23.95,
  "City": "Thai Nguyen"
},
{
  "Date": "2024-05-18",
  "Temperature": 24.23,
  "City": "Thai Nguyen"
},
{
  "Date": "2024-05-19",
  "Temperature": 22.84,
  "City": "Thai Nguyen"
}
}
```

Sử dụng node- red và dán đường link vào http response

Edit http request node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖼

☰ Method

GET

▼

🌐 URL

http://127.0.0.1:8000/weather

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

📄

☐ Enabled

Tiếp đó viết 1
file để lưu dữ
liệu vào
database SQL

```
save_data.py x
1 import requests
2 import pyodbc
3 import logging
4
5 # Cấu hình logging
6 logging.basicConfig(level=logging.INFO)
7 logger = logging.getLogger(__name__)
8
9 def fetch_weather_data(url):
10     try:
11         response = requests.get(url)
12         response.raise_for_status()
13         data = response.json()
14         logger.info("Successfully fetched data from the URL.")
15         return data
16     except requests.exceptions.RequestException as e:
17         logger.error(f"Error fetching data from {url}: {e}")
18         return None
19
20 usage new *
```

Kết quả lưu trong
database SQL

SQLQuery3.sql - TH....THAYCOP (sa (53))

SQLQuery2.sql - TH....

	city		date	temp
	Thai Nguyen	...	2024-05-14	30.88
	Thai Nguyen	...	2024-05-15	25.75
	Thai Nguyen	...	2024-05-16	23.61
	Thai Nguyen	...	2024-05-17	23.78
	Thai Nguyen	...	2024-05-18	24.59
	Thai Nguyen	...	2024-05-19	23.49
►*	NULL		NULL	NULL

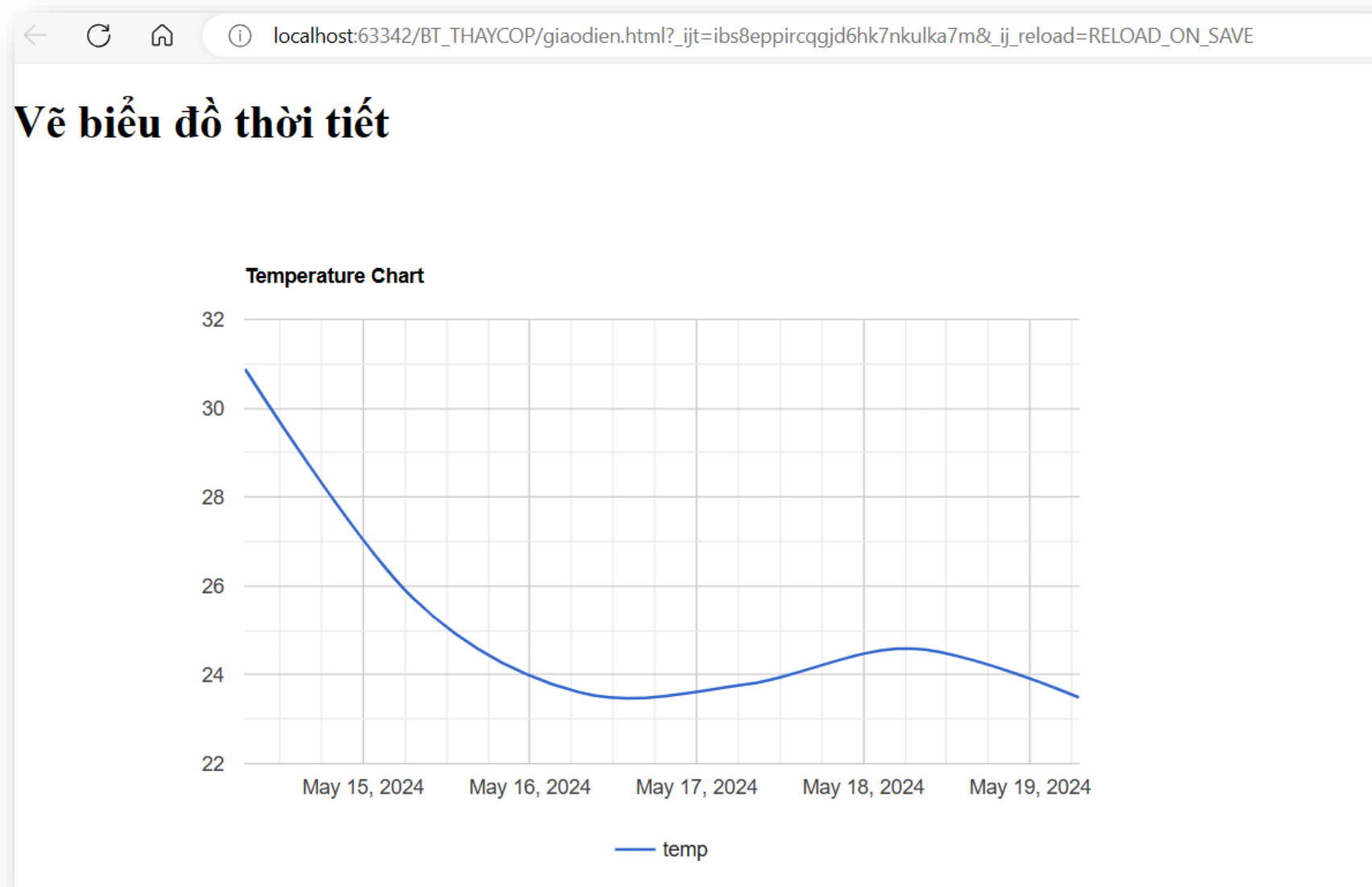
Tạo file để tải dữ liệu nhiệt độ từ một URL JSON và vẽ biểu đồ nhiệt độ bằng Google Charts

```
JS bieudo.js x
1 // Hàm để tải dữ liệu từ URL JSON
2 google.charts.load('current', {'packages':['corechart']});
3 google.charts.setOnLoadCallback(drawCharts);
4 1+ usages THAO
5 function drawCharts() :void {
6     ⚡ drawTemperatureChart();
7 }
8 1+ usages THAO
9 function loadData(url, callback) :void {
10     fetch(url) Promise<Response>
11     .then(response :Response => response.json()) Promise<any>
12     .then(data => callback(data)) Promise<any>
13     .catch(error => console.error('Error loading data:', error));
14 }
15 // Hàm để vẽ biểu đồ nhiệt độ
16 1+ usages THAO
17 function drawTemperatureChart() :void {
18     loadData( url: 'http://127.0.0.1:1880/get-temp', callback: function(data) :void {
19         // Tạo mảng dữ liệu cho biểu đồ
20         var chartData :any[] = [];
21         chartData.push(['date', 'temp']); // Thêm tiêu đề cột
22
23         // Lặp qua dữ liệu JSON và thêm vào mảng dữ liệu
24         data.forEach(function(entry) :void {
25             chartData.push([new Date(entry.date), entry.temp]);
26         });
```

Tạo một trang web để hiển thị biểu đồ thời tiết bằng Google Charts

```
<> giaodien.html x
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Charts</title>
7
8      <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
9      <script src="bieudo.js"></script>
10 </head>
11 <body>
12     <h1>Vẽ biểu đồ thời tiết</h1>
13     <!-- Đây là nơi để hiển thị các biểu đồ -->
14     <div id="temperature_chart" style="width: 900px; height: 500px;"></div>
15
16 </body>
17 </html>
18
```

Kết quả



Toàn bộ dữ liệu bài làm em để ở trong:

⇒ https://github.com/Thao1406/BT_THAYCOP_14-5-2024

Do kiến thức có hạn nên bài làm còn khá đơn giản,
mong thầy chiếu cố ạ!!!

