

# Quá trình hoàn thiện bài tập.

Họ và tên: Nguyễn Đức Mạnh

Mssv: K205480106038

Lớp : 56KMT.01

Môn: Lập trình Python

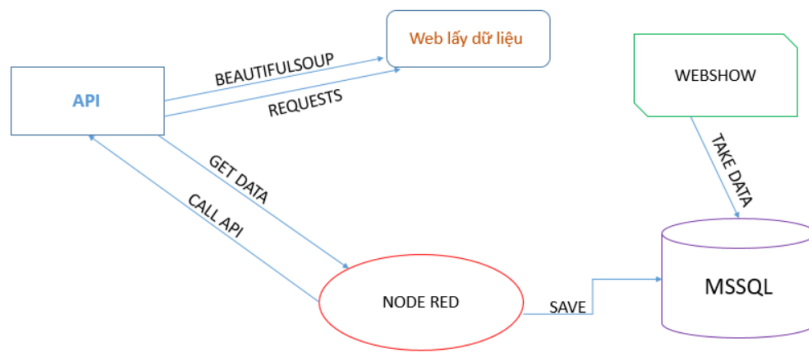
Đề bài Theo dõi giá xe hơi Vinfast

## 1. Quá trình cài đặt và môi trường

- Firstly, sử dụng anaconda để chạy pycharm để sử dụng không gian riêng quản lý bởi anaconda
- Secondly, trong terminal của workspace cài đặt nodeJS và fastapi  
<https://fastapi.tiangolo.com/#installation>  
<https://nodejs.org/en/download/current>
- Cài đặt node-red: yêu cầu cài nodejs trước để kích hoạt npm sau đó:  
npm install -g --unsafe-perm node-red
- Cài SQL Server 2022 :
  1. Truy cập vào đường link: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>.
  2. Tìm đến Express edition of **SQL Server 2022** và chọn **Download** now, chọn Open file.
  3. Chọn Basic -> Accept -> Chọn Browser lưu trữ -> Install.
  4. Sau khi **tải** về thành công, ấn Install SSMS.
  5. Sau khi cửa sổ **Download** SSMS, ấn vào Link **Download** SSMS.
  6. Sau khi **Download** thành công, ấn Install.
  7. Sau khi cài đặt mở Microsoft **SQL Server** Management Studio 19, chọn Connect.

## 2. Làm bài

- Sơ đồ hoạt động



Quá trình :

- Tạo api fast api python lấy data trên web data
- Dùng node red get data api, lưu data về mssql
- Dùng python để gọi dữ liệu từ mssql ra đưa lên web
  - Input: Thông tin từ [https://vinfastauto.com/vn\\_vi/bang-gia-xe-vinfast-moi-nhat](https://vinfastauto.com/vn_vi/bang-gia-xe-vinfast-moi-nhat)
  - Output: Biểu đồ thông tin
  - Quá trình tìm hiểu và nghiên cứu có thời gian khá ngắn nên không hiểu chi tiết về từng code và công dụng nên phải hỏi chatGPT khá nhiều.

Tạo cơ sở dữ liệu

- Tạo ra 1 bảng có tên là Cars, trong bảng gồm có CarID, Brand, Model, Year, Price

```

USE [CarDatabase]
GO
/***** Object: Table [dbo].[Cars]    Script Date: 05/17/2024 6:27:51 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Cars](
    [CarID] [int] IDENTITY(1,1) NOT NULL,
    [Brand] [nvarchar](50) NULL,
    [Model] [nvarchar](50) NULL,
    [Year] [datetime] NULL,
    [Price] [float] NULL,
    PRIMARY KEY CLUSTERED
    (
        [CarID] ASC
    )

```

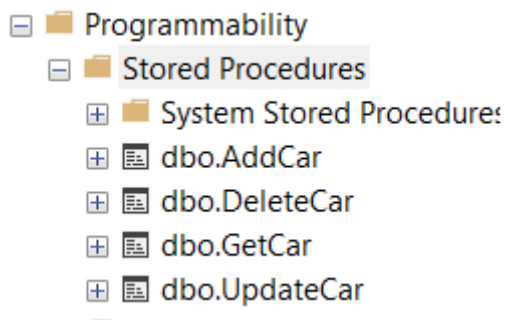
```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY]  
) ON [PRIMARY]
```

## Tạo Stored Procedures để kết nối với node red

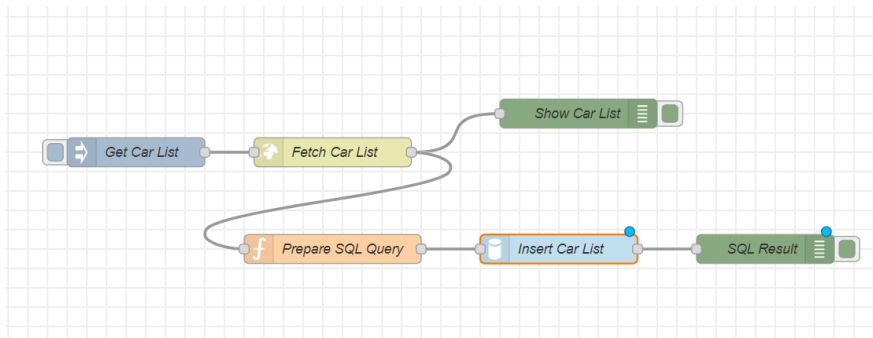
- Nó cung cấp một cách để thực thi các tập hợp câu lệnh SQL đã được biên dịch trước, giúp tăng hiệu suất và dễ quản lý các thao tác phức tạp với cơ sở dữ liệu

```
use CarDatabase  
go  
CREATE PROCEDURE AddCar  
    @Brand NVARCHAR(50),  
    @Model NVARCHAR(50),  
    @Year DateTime,  
    @Price FLOAT  
AS  
BEGIN  
    INSERT INTO Cars (Brand, Model, Year, Price)  
    VALUES (@Brand, @Model, @Year, @Price);  
END;
```

## Kết quả hiển thị

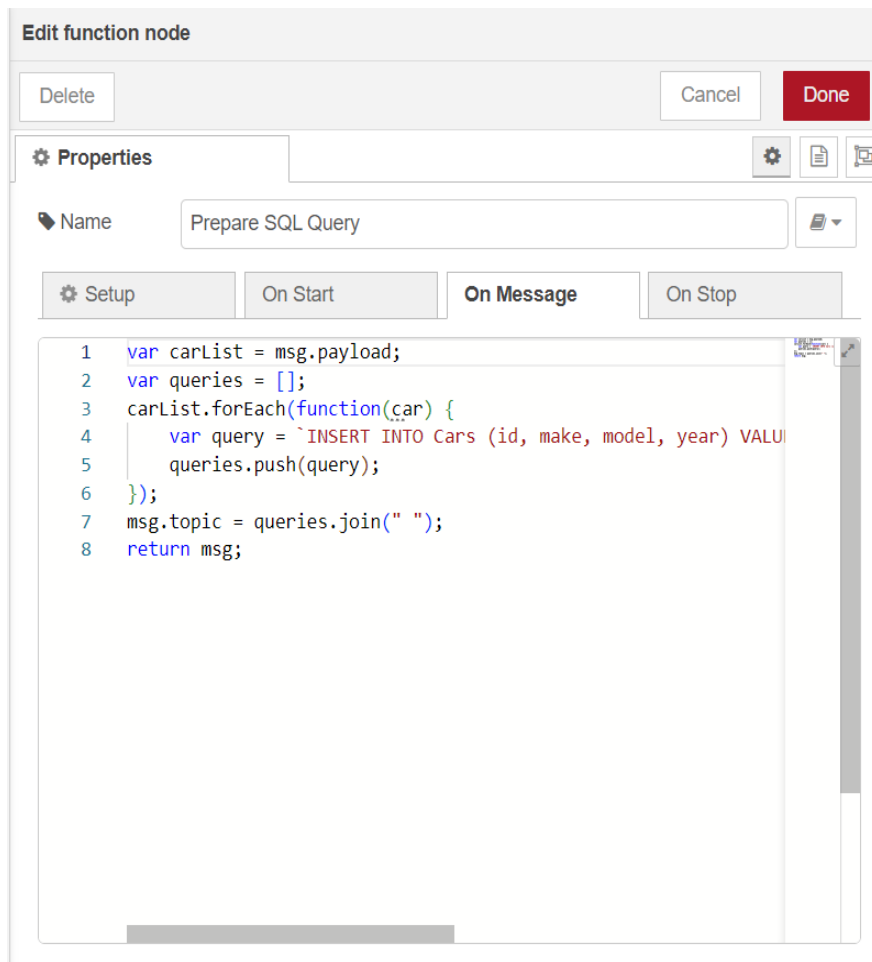


## Tạo Node-red



Tiếp theo em sẽ sử dụng node-red và dán đường link vào http response

- Tạo các câu lệnh SQL từ dữ liệu đầu vào: Node này nhận một danh sách các đối tượng xe từ msg.payload, sau đó tạo các câu lệnh SQL để chèn dữ liệu vào bảng "Cars".
- Chuẩn bị cho việc chèn dữ liệu vào cơ sở dữ liệu: Các câu lệnh SQL được chuẩn bị và gán vào msg.topic, sẵn sàng để được thực thi bởi một node khác trong luồng Node-RED, chẳng hạn như một node cơ sở dữ liệu.



- Kết nối sql server với MSSQL node trên Node RED

**Edit MSSQL node**

Delete Cancel Done

**Properties**

Connection 127.0.0.1.CarDatabase

Name Insert Car List

Query mode Execute Procedure

Query msg. payload

Parameters msg. payload

Parse Mustache ☒

Output property msg. payload

Output type Original output

Error Handling Throw error

**Edit MSSQL node > Edit MSSQL-CN node**

Delete Cancel Update

**Properties**

Name MSSQL Connection

Server 127.0.0.1

Port 1433

Username sa

Password .....

Domain

Database CarDatabase

TDS Version 7\_4 (SQL Server 2012 ~ 2022)

Use Encryption? ☐

SQL Databases hosted on Azure will need this checked.

Trust Certificate? ☐

If unchecked, SQL Server will try to validate the server SSL certificate and will terminate the connection if validation fails

Enabled 1 On all flows

## Lấy dữ liệu api

```
@app.get("/Cars", response_class=HTMLResponse)
def get_data() -> List[Dict[str, str]]:
    url = 'https://vinfastauto.com/vn_vi/bang-gia-xe-vinfast-moi-nhat'
    try:
        response = requests.get(url)
        response.raise_for_status() # Kiểm tra trạng thái của yêu cầu HTTP
    except requests.RequestException as e:
        raise HTTPException(status_code=500, detail=f"Lỗi khi truy cập URL: {e}")

    soup = BeautifulSoup(response.content, 'html.parser')

    # Trích xuất dữ liệu từ trang web
    data = []
    table = soup.find('table')
    if not table:
        raise HTTPException(status_code=404, detail="Không tìm thấy bảng dữ liệu")

    rows = table.find_all('tr')[1:]
    for row in rows:
        cols = row.find_all('td')
        if len(cols) > 2:
            car_info = {
```

```

        'Brand': cols[0].get_text(strip=True),
        'Model': cols[1].get_text(strip=True),
        'Year': cols[2].get_text(strip=True),
        'Price': cols[2].get_text(strip=True)
    }
    data.append(car_info)
print(data)
return data
return HTMLResponse(content=html_content)

```

- Mảng Json lấy đc từ web data

```

1  [
2      {
3          "Brand": "Vinfast",
4          "Model": "VF3",
5          "Year": 2024,
6          "Price": 315000000
7      },
8      {
9          "Brand": "Vinfast",
10         "Model": "VFe34",
11         "Year": 2021,
12         "Price": 700000000
13     },
14     {
15         "Brand": "Vinfast",
16         "Model": "VF5",
17         "Year": 2023,
18         "Price": 479000000
19     },
20     {
21         "Brand": "Vinfast",
22         "Model": "VF6",
23         "Year": 2023,
24         "Price": 776000000
25     },
26     {
27         "Brand": "Vinfast",
28         "Model": "VF7",
29         "Year": 2023,
30         "Price": 1200000000
31     },
32     {
33         "Brand": "Vinfast",
34         "Model": "VF8",
35         "Year": 2023,
36         "Price": 1300000000
37     },
38     {
39         "Brand": "Vinfast",
40         "Model": "VF9",
41         "Year": 2023,
42         "Price": 1700000000
43     }
44 ]

```

Xuất ra kết quả

|   | CarID | Brand   | Model | Year | Price      |
|---|-------|---------|-------|------|------------|
| 1 | 1     | Vinfast | VF3   | 2024 | 315000000  |
| 2 | 2     | Vinfast | VFe34 | 2021 | 700000000  |
| 3 | 3     | Vinfast | VF5   | 2023 | 479000000  |
| 4 | 4     | Vinfast | VF6   | 2023 | 776000000  |
| 5 | 5     | Vinfast | VF7   | 2023 | 1200000000 |
| 6 | 6     | Vinfast | VF8   | 2023 | 1300000000 |
| 7 | 7     | Vinfast | VF9   | 2023 | 1700000000 |

Kết quả sau khi chạy

.

Car List

| Brand   | Model | Year | Price        |
|---------|-------|------|--------------|
| Vinfast | VF3   | 2024 | 315000000.0  |
| Vinfast | VFe34 | 2021 | 700000000.0  |
| Vinfast | VF5   | 2023 | 479000000.0  |
| Vinfast | VF6   | 2023 | 776000000.0  |
| Vinfast | VF7   | 2023 | 1200000000.0 |
| Vinfast | VF8   | 2023 | 1300000000.0 |
| Vinfast | VF9   | 2023 | 1700000000.0 |

