

Hướng dẫn sử dụng Rest API SQLAlchemy với Flask – Phần I

Giáo viên: Nguyễn Hùng Cường

Bước 1: Cài đặt các thư viện cần thiết

Mở một trình soạn thảo, ở đây ta sử dụng PyCharm IDE. Sau đó ta tạo một Project Flask. Chú ý nhớ mở Terminal, rồi gõ lệnh `pip install flask` để cài đặt Flask Framework.

Tiếp theo, ta gõ lệnh sau để cài đặt thư viện Flask-RESTful, để có thể tạo và gọi Rest Api trong ứng dụng.

```
Terminal: Local x + v
PS D:\ademobaigiang2023\demoflask\flaskquanlynhanvien> pip install Flask-RESTful
Collecting Flask-RESTful
  Downloading Flask-RESTful-0.3.10-py2.py3-none-any.whl (26 kB)
Collecting six>=1.3.0
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Requirement already satisfied: Flask>=0.8 in d:\ademobaigiang2023\demoflask\flaskquanlynhanvien\venv\lib\site-packages (from Flask-RESTful) (2.0.1)
Collecting pytz
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
Requirement already satisfied: colorama in d:\ademobaigiang2023\demoflask\flaskquanlynhanvien\venv\lib\site-packages (from Flask-RESTful) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in d:\ademobaigiang2023\demoflask\flaskquanlynhanvien\venv\lib\site-packages (from Flask-RESTful) (2.1.3)
Installing collected packages: six, pytz, aniso8601, Flask-RESTful
Successfully installed Flask-RESTful-0.3.10 aniso8601-9.0.1 pytz-2023.3.post1 six-1.16.0
WARNING: You are using pip version 21.3.1; however, version 23.3.1 is available.
You should consider upgrading via the 'D:\ademobaigiang2023\demoflask\flaskquanlynhanvien\venv\Scripts\python.exe -m pip install --upgrade pip'
PS D:\ademobaigiang2023\demoflask\flaskquanlynhanvien>
```

Tiếp theo, ta gõ lệnh sau để cài đặt thư viện Flask-SQLAlchemy, để có thể sử dụng cơ chế ORM trong ứng dụng này.

```
PS D:\ademobaigiang2023\demoflask\flaskquanlynhanvien> pip install Flask-SQLAlchemy
Collecting Flask-SQLAlchemy
  Downloading flask_sqlalchemy-3.1.1-py3-none-any.whl (25 kB)
Requirement already satisfied: flask>=2.2.5 in d:\ademobaigiang2023\demoflask\flaskquanlynhanvien\venv\lib\site-packages (from Flask-SQLAlchemy) (3.0.0)
Collecting sqlalchemy>=2.0.16
  Downloading SQLAlchemy-2.0.23-cp310-cp310-win_amd64.whl (2.1 MB)
  2.1 MB 3.2 MB/s
Collecting greenlet<=0.4.17
  Downloading greenlet-3.0.2-cp310-cp310-win_amd64.whl (288 kB)
  288 kB 6.4 MB/s
Requirement already satisfied: colorama in d:\ademobaigiang2023\demoflask\flaskquanlynhanvien\venv\lib\site-packages (from click>=8.1.3->flask>=2.2.5->Flask-SQLAlchemy) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in d:\ademobaigiang2023\demoflask\flaskquanlynhanvien\venv\lib\site-packages (from Jinja2>=3.1.2->flask>=2.2.5->Flask-SQLAlchemy) (2.1.3)
Installing collected packages: typing-extensions, greenlet, sqlalchemy, Flask-SQLAlchemy
Successfully installed Flask-SQLAlchemy-3.1.1 greenlet-3.0.2 sqlalchemy-2.0.23 typing-extensions-4.9.0
WARNING: You are using pip version 21.3.1; however, version 23.3.1 is available.
You should consider upgrading via the 'D:\ademobaigiang2023\demoflask\flaskquanlynhanvien\venv\Scripts\python.exe -m pip install --upgrade pip'
PS D:\ademobaigiang2023\demoflask\flaskquanlynhanvien>
```

Ở trong bài này, ta sẽ tạo một ứng dụng Flask có tạo API để kết nối với CSDL SQLite. Trong bài sử dụng SQLAlchemy để hỗ trợ ORM trong ứng dụng.

Bước 2: Viết mã cho ứng dụng

Tiếp theo, ta cần viết mã cho ứng dụng. Ta tạo một số file mã nguồn Python, đầu tiên là file config.py với khai báo url đến database như sau.

```
config.py x
1  SQLAlchemy_DATABASE_URI = 'sqlite:///quanlynhanvien.db'
2
```

Sau đó, ta định nghĩa file models.py, file này sẽ định nghĩa class NhanVien để mapping với bảng nhanvien trong database.

```
models.py x
1  from flask_sqlalchemy import SQLAlchemy
2
3  db = SQLAlchemy()
4
5  # định nghĩa class để mapping với bảng nhanvien trong CSDL
6  class NhanVien(db.Model):
7      ma = db.Column(db.Integer, primary_key=True)
8      # khai báo nullable=False để cột bắt buộc phải nhập
9      hoten = db.Column(db.String(50), nullable=False)
10     tuoi = db.Column(db.Integer)
11     diachi = db.Column(db.String(50))
12
13     def __repr__(self):
14         # Method này cho phép hiển thị thông tin chi tiết về đối tượng
15         return f'Nhân viên {self.ma} : {self.hoten} : {self.tuoi} : {self.diachi}'
```

Sau đó, ta định nghĩa file services.py, file này chứa định nghĩa class NhanVienManager gồm 2 api cho 2 method là post và get, để thêm mới nhân viên và trả về danh sách nhân viên từ CSDL như sau.

```

services.py x
1  from flask_restful import Resource
2  from flask import request
3  from models import NhanVien, db
4
5  # định nghĩa class cung cấp các method cho phép gọi API từ frontend
6  class NhanVienManager(Resource):
7      def get(self):
8          # lấy về tất cả nhân viên từ database.
9          employees = NhanVien.query.all()
10         # Chuyển đổi danh sách nhân viên thành JSON và trả về response.
11         listemployee = [{'ma': employee.ma, 'hoten': employee.hoten, 'tuoi': employee.tuoi, 'diachi': employee.diachi}
12                         for employee in employees]
13         return {'listemployee': listemployee}

```

```

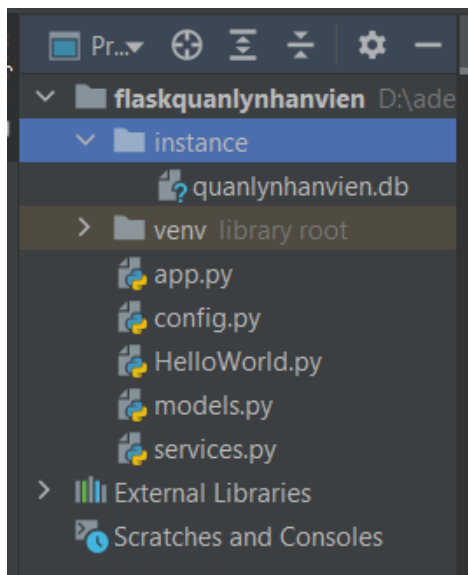
services.py x
15  def post(self):
16      # Lấy JSON data từ request.
17      response_data = request.get_json()
18      # Kiểm tra xem dữ liệu có hợp lệ hay không.
19      if not response_data:
20          return {'message': 'No input data provided'}, 400
21      hoten = response_data.get('hoten')
22      if not hoten:
23          return {'message': 'Họ tên nhân viên không được bỏ trống !'}, 400
24      # Thêm nhân viên mới vào database.
25      new_employee = NhanVien(hoten='Đặng Thu Thảo', tuoi=30, diachi='Settle')
26      db.session.add(new_employee)
27      # Ghi dữ liệu vào database.
28      db.session.commit()
29      # Trả về một message cho người dùng.
30      return {'message': 'Đã thêm nhân viên mới', 'nhanvien': {'ma': new_employee.ma,
31                                                             'hoten': new_employee.hoten,
32                                                             'tuoi': new_employee.tuoi,
33                                                             'diachi': new_employee.diachi}}

```

Cuối cùng, ta định nghĩa file app.py, đây là file thực thi ứng dụng. Trong file này ta sẽ khởi tạo database, tạo Api để client có thể gọi. Chú ý Api đã được khai báo với url là /nhanvien.

```
app.py x
1  from flask import Flask
2      from flask_restful import Api
3      from models import db
4      import config
5  from services import NhanVienManager
6
7      # Khởi tạo Flask application và Flask-RESTful API manager.
8      app = Flask(__name__)
9      app.config.from_object(config)
10     # Khởi tạo đối tượng Flask-SQLAlchemy.
11     db.init_app(app)
12     # Tạo đối tượng Flask-RESTful API manager.
13     api = Api(app)
14     # Tạo endpoints.
15     api.add_resource(NhanVienManager, '/nhanvien')
16
17  if __name__ == '__main__':
18      # Tạo database tables.
19      with app.app_context():
20          db.create_all()
21      # Start Flask development web server.
22      app.run(debug=True)
```

Dưới đây là cấu trúc thư mục của project Flask mà ta vừa tạo ở bước trên.



Tiếp theo, ta thực thi ứng dụng, đầu tiên ta sẽ thấy IDE hiển thị thông báo như hình bên dưới.

```
D:\ademobaigiang2023\demoflask\flaskquanlynhanvien\venv\Scripts\python.exe D:/ademobaigiang2023/demoflask/flaskquanlynhanvien/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 995-036-031
```

Sau đó, gõ url đã được khai báo, ta sẽ thấy Api Get đã trả về một danh sách rỗng, vì ta chưa chèn đối tượng nhân viên nào vào bảng, nên Api Get chỉ trả về list rỗng.

