BUỔI 3: API VÀ FASTAPI

MỤC TIÊU:

- Làm quen với API
- Thực hiện một số thao tác cơ bản với MySQL thông qua FastAPI

CHUẨN BỊ:

- Requirements: python 3.9+
- Installation: pip install fastapi
- Postman: https://www.postman.com/downloads/

I. API

1. Giới thiệu về API

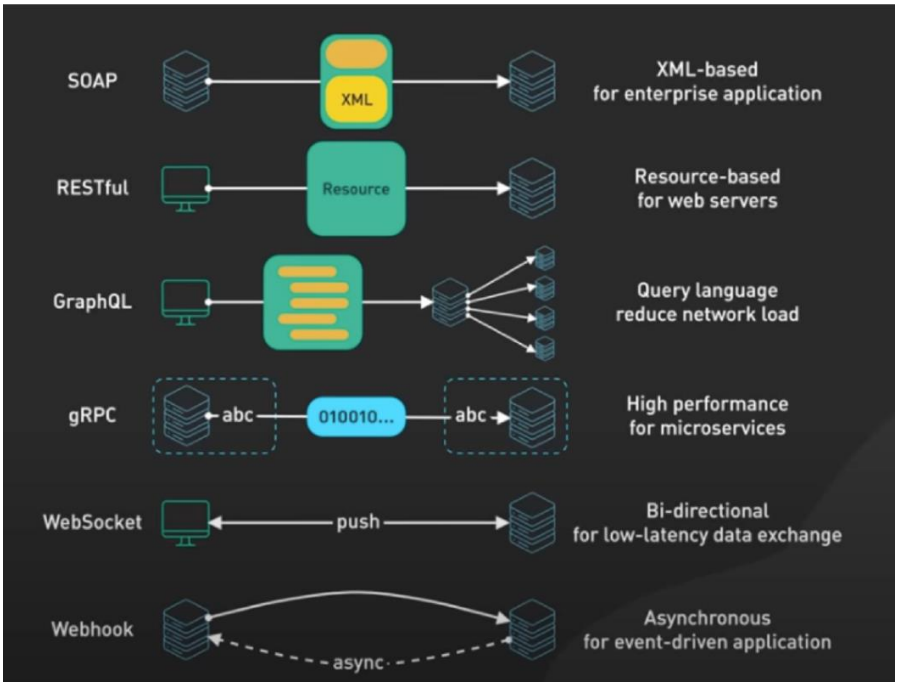
Là viết tắt của "Application Programming Interface" (Giao diện lập trình ứng dụng) đề cập đến một tập hợp các quy tắc và giao thức mà cho phép các phần mềm ứng dụng giao tiếp với nhau.

API giúp các phần mềm và ứng dụng khác nhau trao đổi dữ liệu và tương tác với nhau một cách hiệu quả.

Dưới đây là một số điểm quan trọng về tại sao API quan trọng trong lập trình:

- Tích hợp ứng dụng: API cho phép kết nối ứng dụng của mình với các dịch vụ và ứng dụng khác, giúp tích hợp các chức năng và dữ liệu từ nhiều nguồn khác nhau.
- Tái sử dụng mã nguồn
- Phân chia công việc: API cho phép nhiều nhóm lập trình làm việc độc lập nhau trên các phần của dự án và sau đó kết hợp chúng thông qua API
- Đảm bảo tính nhất quán: API định rõ cách các ứng dụng khác nhau nên tương tác với nhau.
- Bảo mật và quản lý quyền truy cập: API cho phép quản lý quyền truy cập vào các tài nguyên và dữ liệu.

2. Phân biệt giữa RESTful API, SOAP API, GraphQL, và các loại API khác



3. Các khái niệm cơ bản về HTTP, URL

a. HTTP (Hypertext Transfer Protocol):

Là một giao thức mạng dùng để truyền tải dữ liệu giữa máy tính trình duyệt (hoặc ứng dụng) và máy chủ web.

Nó hoạt động dựa trên mô hình yêu cầu/đáp ứng (request/response), trong đó trình duyệt gửi yêu cầu đến máy chủ và máy chủ trả lời với dữ liệu (thông qua các phương thức như GET, POST, PUT, DELETE).

b. URL (Uniform Resource Locator):

URL là một chuỗi ký tự định vị tài nguyên trên mạng. Gồm các thành phần:

- Giao thức (Protocol): Ví dụ: http, https, ftp.
- Tên miền (Domain Name): Ví dụ: www.example.com.
- Đường dẫn (Path): Định vị tài nguyên cụ thể trên máy chủ.
- Tham số (Query Parameters): Thông tin bổ sung được gửi cùng yêu cầu.
- Phân đoạn (Fragment): Chỉ định vị trí cụ thể trong tài liệu nếu là tài liệu đa phần

4. Các phương thức cơ bản trong API (CRUD)

- **GET**: Phương thức này được sử dụng để truy vấn thông tin từ tài nguyên. Nó không làm thay đổi dữ liệu, chỉ đọc dữ liệu hiện có.
- **POST**: Được sử dụng để tạo mới một tài nguyên. Khi gửi một yêu cầu POST, đang gửi dữ liệu để tạo một tài nguyên mới.

- **PUT**: Sử dụng để cập nhật hoặc thay đổi một tài nguyên đã tồn tại hoàn toàn. cung cấp tất cả thông tin của tài nguyên trong yêu cầu PUT.
- **DELETE**: Được sử dụng để xóa một tài nguyên khỏi hệ thống.

II. FastAPI

Là một web framework hiện đại, nhanh chóng (hiệu suất cao) trong xây dựng API bằng Python 3.7+ dựa trên gợi ý loại Python tiêu chuẩn.

- **Fast**: Hiệu suất rất cao, ngang bằng với NodeJS và Go (nhờ Starlette và Pydantic). Một trong những framework Python nhanh nhất hiện có .
- **Fast to code**: Tăng tốc độ phát triển các tính năng khoảng 200% đến 300%.
- **Fewer bugs**: Giảm khoảng 40% lỗi do con người (nhà phát triển) gây ra.
- **Intuitive** : Hỗ trợ biên tập viên tuyệt vời. Hoàn thành ở khắp mọi nơi. Ít thời gian gỡ lỗi hơn.
- **Easy**: Được thiết kế để dễ sử dụng và học hỏi. Ít thời gian đọc tài liệu.
- **Short**: Giảm thiểu việc trùng lặp mã. Nhiều tính năng từ mỗi khai báo tham số. Ít lỗi hơn.
- **Robust**: Nhận mã sẵn sàng sản xuất. Với tài liệu tương tác tự động.
- **Standards-based**: Dựa trên (và hoàn toàn tương thích với) các tiêu chuẩn mở cho API:API mở(trước đây gọi là Swagger) và Lược đồ JSON.

Đầu tiên chúng ta sẽ khởi tạo một tệp chương trình FastAPI có cú pháp đơn giản có tên “main.py” gồm nội dung như sau:

```
from fastapi import FastAPI

app = FastAPI()


@app.get("/")
async def root():
    return {"message": "Hello World"}
```

Khởi chạy bằng lệnh uvicorn main:app --reload trên terminal sau đó mở trình duyệt Chrome hoặc bất cứ trình duyệt nào, gõ địa chỉ http://127.0.0.1:8000/

Kết quả sẽ trả về:

{“message”: “Hello World”}

1. **Tham số theo địa chỉ**

Cú pháp truyền tham số theo địa chỉ như sau:

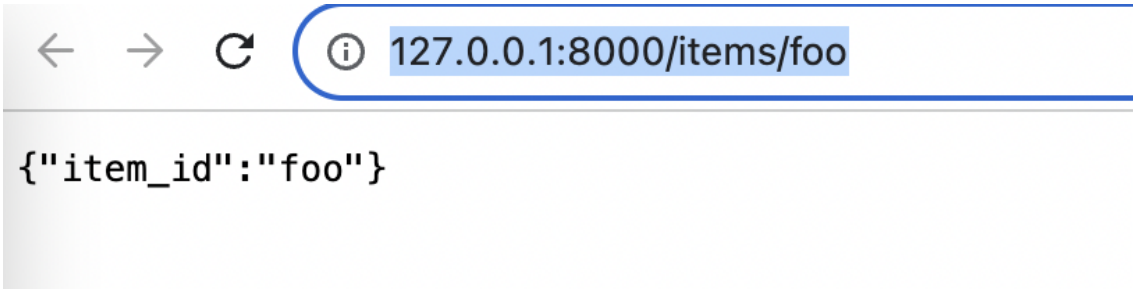
URL/<đường dẫn nếu có>/<tham số truyền vào>

Ví dụ:

http://127.0.0.1:8000/items/<giá trị truyền vào>

```python
from fastapi import FastAPI

app = FastAPI()


@app.get("/items/{item_id}")
async def read_item(item_id):
    return {"item_id": item_id}
```

Đi đến http://127.0.0.1:8000/items/foo trong trình duyệt:



2. **Tham số theo truy vấn**

Tham số theo truy vấn (query parameter) là một cách để truyền thông tin đến một trang web hoặc API thông qua URL. Đây là cách để gửi dữ liệu từ máy khách (client) đến máy chủ (server) để máy chủ có thể xử lý dữ liệu đó.

Tham số theo truy vấn thường được thêm vào URL sau dấu hỏi chấm (?), và có thể bao gồm nhiều cặp tên giá trị. Có các kiểu tham số sau:

- Tham số mặc định: Là các tham số có sẵn, nếu không truyền giá trị vào tham số đó thì sẽ luôn có giá trị mặc định.
- Tham số bắt buộc: Là tham số bắt buộc phải được truyền giá trị vào.
- Tham số tuỳ chọn: có thể chọn kiểu dữ liệu truyền vào bất kỳ.

Cú pháp truyền tham số theo địa chỉ như sau:

URL/<đường dẫn nếu có>?<tham số truy vấn>=<giá trị truy vấn>

Ví dụ:

http://127.0.0.1:8000/items/soap?needy=yes

Ngoài ra có thể điền thêm các tham số truy vấn khác bằng dấu "&":
http://127.0.0.1:8000/items/soap?needy=yes&skip=5&limit=10

```python
from fastapi import FastAPI

app = FastAPI()


@app.get("/items/soap")
async def read_user_item(needy: str, skip: int = 0, limit: int | None = None):
    item = {"needy": needy, "skip": skip, "limit": limit}
    return item
```

- **needy**: một tham số bắt buộc với kiểu str.
- **skip**: một int giá trị mặc định là 0.
- **limit**: giá trị truyền vào có thể là int hoặc None, mặc định là None

Nếu mở trình duyệt http://127.0.0.1:8000/items/soap sẽ nhận được lỗi là:

{"detail":[{"loc":["query","needy"],"msg":"field required","type":"value_error.missing"}]}

do **needy** là tham số bắt buộc, cần đặt trong URL:
http://127.0.0.1:8000/items/soap?needy=yes

3. Request Body

Khi cần gửi dữ liệu từ một máy khác (ví dụ, một trình duyệt) thông qua API, có thể gửi dữ liệu đó dưới dạng Request Body.

```python
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel

class Item(BaseModel):
    name: str
    description: str | None = None
    price: float
    tax: float | None = None

app = FastAPI()

items = []
```

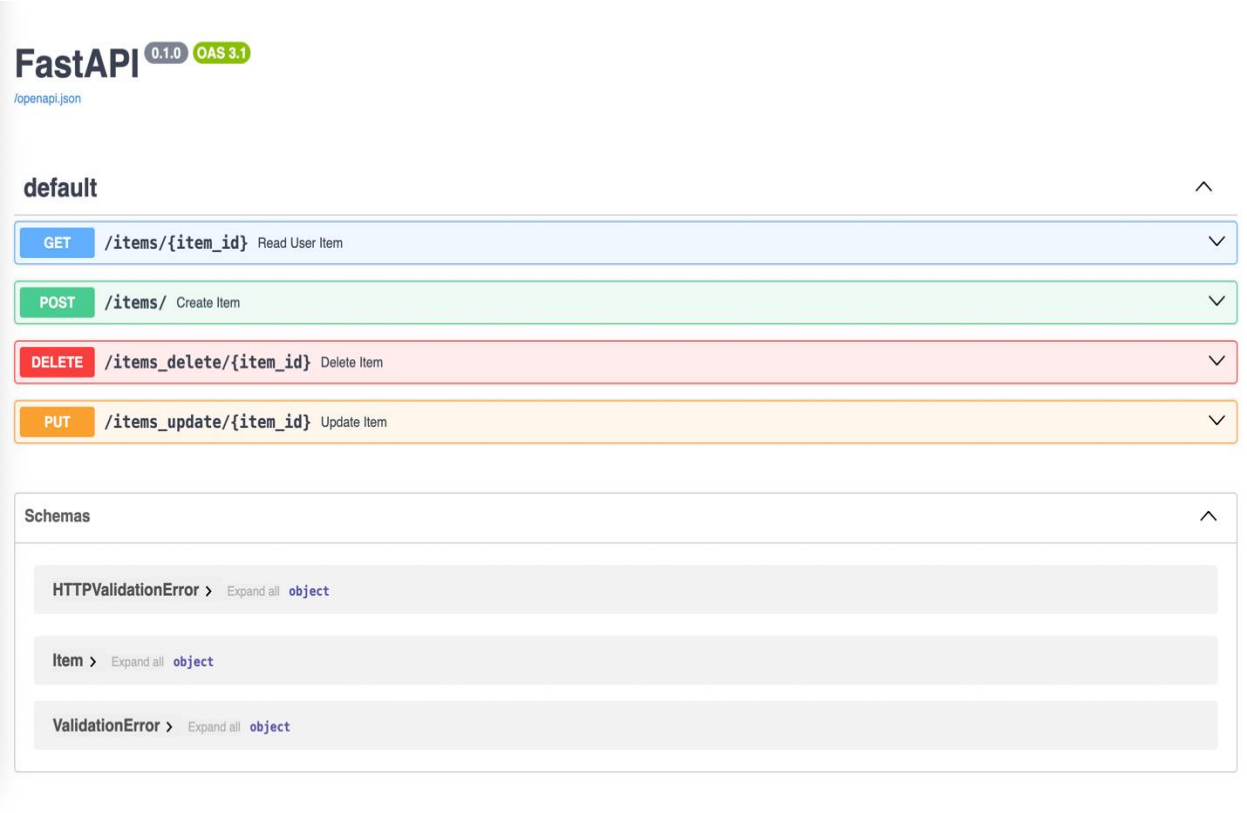
```python
# GET: Lấy danh sách item
@app.get("/items/")
async def list_items():
    return items

# POST: Tạo item mới
@app.post("/items/")
async def create_item(item: Item):
    items.append(item)
    return {"message": "Item created successfully", "item": item}

# PUT: Cập nhật thông tin item
@app.put("/items_update/{item_id}")
async def update_item(item_id: int, updated_item: Item):
    if item_id < 0 or item_id >= len(items):
        raise HTTPException(status_code=404, detail="Item not found")

    items[item_id] = updated_item

    return {"message": "Item updated successfully", "item": updated_item}

# DELETE: Xóa item
@app.delete("/items_delete/{item_id}")
async def delete_item(item_id: int):
    if item_id < 0 or item_id >= len(items):
        raise HTTPException(status_code=404, detail="Item not found")

    deleted_item = items.pop(item_id)
    return {"message": f"Item with ID {item_id} has been deleted",
"deleted_item": deleted_item}
```

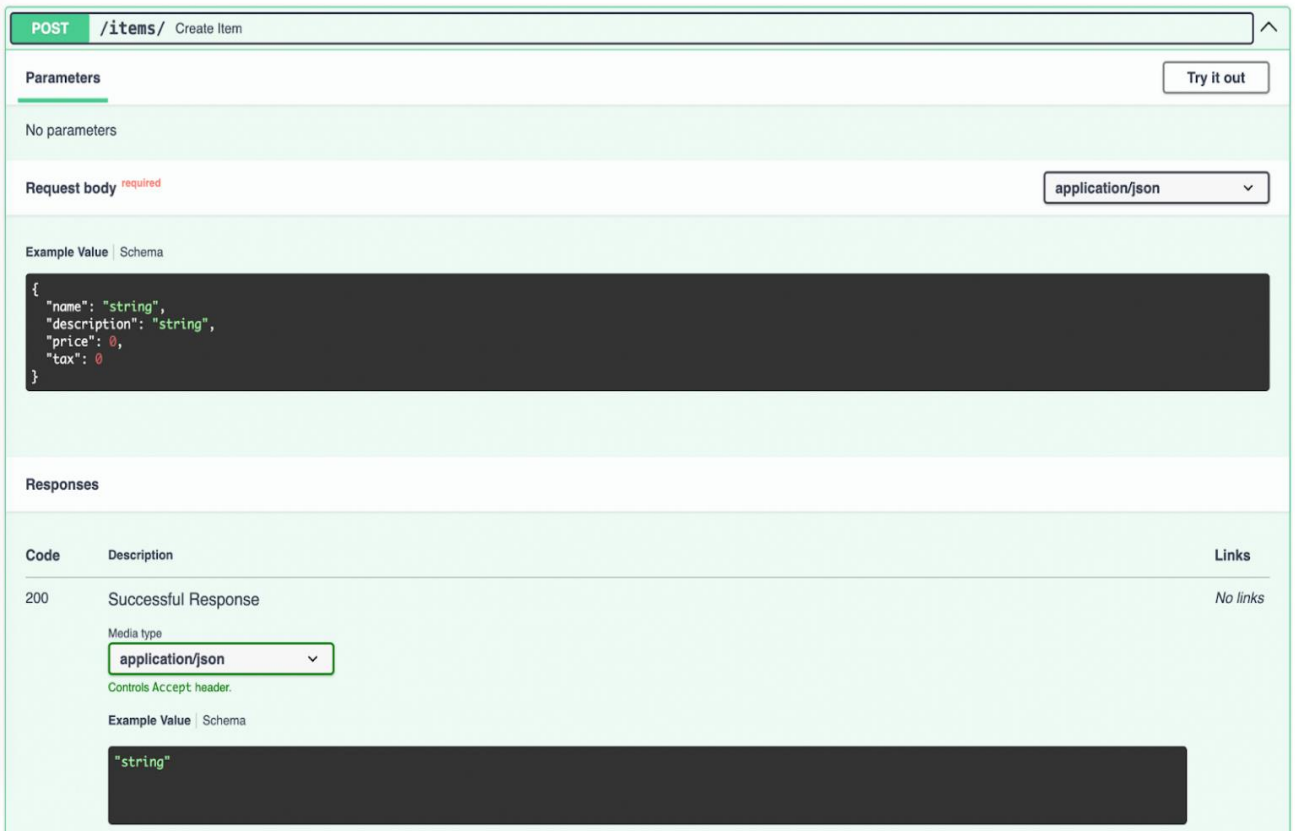
Tuy nhiên, đối với phương pháp này thì chúng ta không thể làm như truyền tham số thông qua trình duyệt mà phải dùng terminal hoặc một số ứng dụng hỗ trợ như Postman, Isomnia… Ở đây trong môn học này chúng ta sẽ sử dụng chính là Postman.

- Tạo một yêu cầu mới:
  - Chọn "+" ở góc trên bên trái.
  - Chọn Request.
- Điền thông tin yêu cầu:
  - Request Type: Chọn POST.
  - URL: http://127.0.0.1:8000/items/.
  - Headers: Không cần thiết.
  - Body: Chọn Raw và ghi JSON data tương tự như sau:

```
{
 "name": "Product 1",
 "description": "This is a sample product",
 "price": 10.5,
 "tax": 1.2
}
```


Để biết thêm về các phương thức hoặc thử tương tác với các API, người dùng có thể mở API Documentation (hay còn gọi là tài liệu API, tài liệu hướng dẫn sử dụng API): http://127.0.0.1:8000/docs





III. Kết nối tới Cơ sở dữ liệu

Trong học phần này, chúng ta sẽ thực hành với Cơ sở dữ liệu MySQL. Đầu tiên, chúng ta sẽ sử dụng XAMPP để kết nối cơ sở dữ liệu MySQL và lấy các thông tin sau:

- Địa chỉ host: chúng ta đang dùng XAMPP nên thông thường địa chỉ sẽ là 127.0.0.1
- Tài khoản: thường tài khoản mặc định sẽ là root@localhost
- Mật khẩu: được thiết lập lúc cài XAMPP
- Database: Tên cơ sở dữ liệu sẽ dùng

Dưới đây là chương trình cơ bản kết nối và truy vấn 1 bảng trong cơ sở dữ liệu:

```
import mysql.connector

host = "localhost"
user = "your_user"
password = "your_password"
database = "your_database"

try:
    conn = mysql.connector.connect(
        host=host,
        user=user,
```

```
        password=password,
        database=database
    )

    cursor = conn.cursor()

    cursor.execute("SELECT * FROM your_table")
    result = cursor.fetchall()

    for row in result:
        print(row)

    cursor.close()
    conn.close()

except Exception as e:
    print(f"Lỗi: {e}")
```

Bài tập: Tạo cơ sở dữ liệu, xây dựng các phương thức cơ bản trong API theo cấu trúc sau:

Yêu cầu:

- Đối với mỗi bảng phải ít nhất 5 bản ghi.
- Viết tài liệu API (API Documentation) về 4 phương thức cơ bản (GET, POST, PUT, DELETE) về cơ sở dữ liệu trên
- Lưu API Documentation theo cú pháp: <mã sinh viên>_buoi3.json

Cấu trúc cơ sở dữ liệu ở trang sau

CƠ SỞ DỮ LIỆU HÀNG KHÔNG

Cho lược đồ cơ sở dữ liệu quan hệ sau:

- CHUYENBAY(MaCB, GaDi, GaDen, DoDai, GioDi, GioDen, ChiPhi) mô tả thông tin về chuyến bay. Mỗi chuyến bay có một mã số duy nhất, đường bay, giờ đi và giờ đến. Thông tin về đường bay được mô tả bởi ga đi, ga đến, độ dài đường bay và chi phí phải trả cho phi công.
- MAYBAY(MaMB, Loai, TamBay) mô tả thông tin về máy bay. Mỗi máy bay có một mã số duy nhất, tên phân loại và tầm bay là khoảng cách xa nhất máy bay có thể bay mà không cần tiếp nhiên liệu. Một máy bay chỉ có thể thực hiện các chuyến bay có độ dài đường bay nhỏ hơn tầm bay của máy bay đó.
- NHANVIEN(MaNV, Ten, Luong) mô tả thông tin về nhân viên phi hành đoàn gồm phi công và tiếp viên. Mỗi nhân viên có một mã số duy nhất, tên và mức lương.
- CHUNGNHAN(MaNV, MaMB) mô tả thông tin về khả năng điều khiển máy bay của phi công. Nếu nhân viên là phi công thì nhân viên đó có chứng chỉ chứng nhận có thể lái một loại máy bay nào đó. Một phi công chỉ có thể lái một chuyến bay nếu như phi công đó được chứng nhận có khả năng lái loại máy bay có thể thực hiện chuyến bay đó.

Mô tả các thuộc tính:

Thuộc tính	Miền xác định
MaCB	char(5)
GaDi	varchar(50)
GaDen	varchar(50)
DoDai	int
GioDi	time
GioDen	time
ChiPhi	int
MaMB	int
Hieu	varchar(50)
TamBay	int
MaNV	char(9)
Ten	varchar(50)
Luong	int