**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**1851050073-Nguyễn Hữu Hoàng Kiệt**

**QUẢN LÝ NHÀ SÁCH**

**TP. HỒ CHÍ MINH,  2023**

**MỤC LỤC**

## Chương 1: Giới Thiệu Đề Tài

## Chương 2: Kiểm Thử Chức Năng

## Chương 3: Kết Quả Kiểm Thử

**Chương 1: Giới Thiệu Đề Tài**

Đối với dự án này, bạn sẽ tạo API RESTful bằng Node và Express, chứa các điểm cuối để thực hiện một số thao tác CRUD trên tài nguyên bạn đã chọn. Hai hoặc ba điểm cuối là đủ. Dữ liệu phải được lưu giữ trong cơ sở dữ liệu SQLite.

Đầy là Rest API về CRUD Pets.

Yêu cầu: Phải cài đặt môi trường Node, SQLite tool, và IDE tùy chọn (VSCode).

***Các bước thực hiện project:***

Dưới đây là danh sách kiểm tra các nhiệm vụ để giúp bạn kết hợp dự án của mình:

1. Tạo tệp `.gitignore`.
2. Cài đặt express, knex@3.1.0, sqlite3.
3. Cài đặt jest, eslint, nodemon, supertest, cross-env làm phần phụ thuộc của nhà phát triển (devDependencies).
4. Định cấu hình jest (thư viện kiểm thử) và eslint (bộ quy tắc coding) bằng `npx <libname> --init`.
5. Tạo một `knexfile.js` với cấu hình "development" và "testing".
6. Tạo tệp `dbConfig.js` chọn cấu hình chính xác bằng cách sử dụng giá trị của `process.env.NODE\_ENV`, nên đặt valueDefault là development được đặt bên trong thư mục data.
7. Tạo tập tin migrations và seeds trong thư mục data.
8. Tập hợp các tập lệnh "start", "server", "rollback", "migrate" và "seed" trong `package.json` của bạn.
9. Tạo tập lệnh "test" trong `package.json` của bạn bằng cách sử dụng cross-env để chèn `NODE\_ENV` của "testing".
10. Tạo một ứng dụng server express cơ bản với một số chức năng truy cập cơ sở dữ liệu và một số điểm cuối (endpoint).
11. Kiểm tra điểm cuối (endpoint) của bạn theo cách thủ công bằng cách sử dụng Postman, HTTPie hoặc tương tự.
12. Kiểm tra điểm cuối (endpoint) của bạn bằng bài kiểm tra báo cáo.

***Các bước thực hiện cài đặt project:***

// Install package.json

cd node-testing2-project

npm install

// Create pets database

npm run migrate

// Add pets data

npm run seed

// Run server

npm run start

// Testing server

npm run test

**Chương 2: Kiểm Thử Chức Năng**

**Các Chức Năng Controllers tại Server:**

1. Phương thức get (“/”) endpoint

Đây là phương thức mặc định đầu tiên khi call API từ phía server.

Kết quả JSON trả về là một đối tượng có key là api

Minh họa:

{ api: "up" }

1. Phương thức get (“/pets”) endpoint

Đây là phương thức thực hiện call API lấy toàn bộ dữ liệu của bảng pets được lưu tại cơ sở dữ liệu SQLite.

Minh họa:

[

{ name: "Cooper", weight: 20, color: "red" },

      { name: "Molly", weight: 40, color: "yellow" },

{ name: "Charlie", weight: 4, color: "purple" },

]

1. Phương thức get (“/pets/:id”) endpoint

Đây là phương thức thực hiện call API lấy ra một đối tượng cụ thể trong bảng pets được chỉ định tìm kiếm theo id

Minh họa: ví dụ id = 1

{

    "id": 1,

    "name": "Cooper",

    "weight": 20,

    "color": "red"

}

1. Phương thức post (“/post”) endpoint

Đây là phương thức thực hiện call API thêm đối tượng mà mình đã cung cấp thông tin (name, weight, color) bắt buộc.

Minh họa;

{ name: "Fluffy", weight: 1, color: "blue" }

1. Phương thức delete (“/pets/:id”) endpoint

Đây là phương thức thực hiện call API xóa đối tượng bên trong cơ sở dữ liệu thông qua id của đối tượng, nếu thực hiện thành công nó sẽ trả lại JSON thông báo.

Minh họa: ví dụ id = 1

{

message: `pet with id 1 removed from database`,

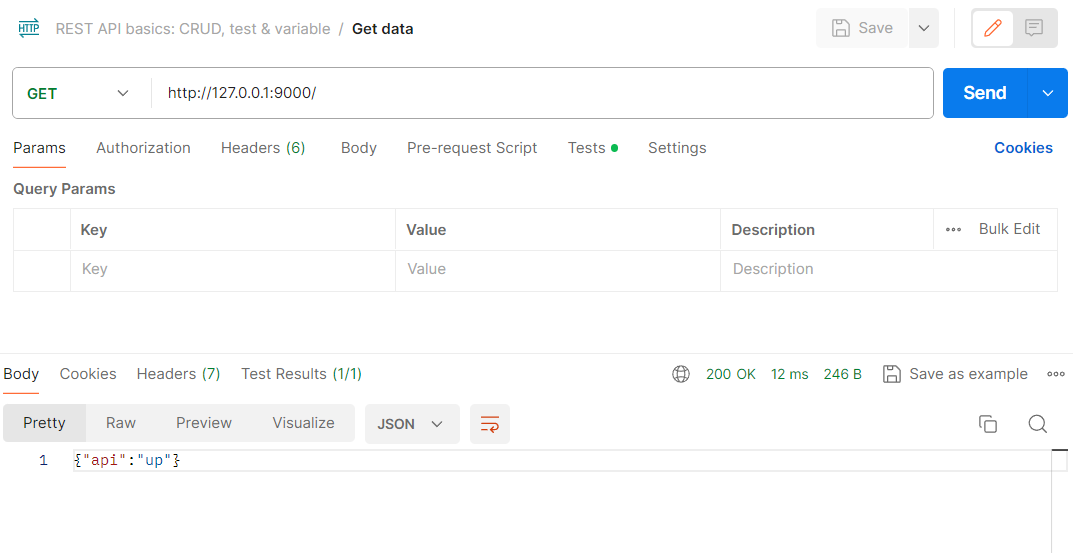
}

**Các Chức Năng Model tại Pets:**

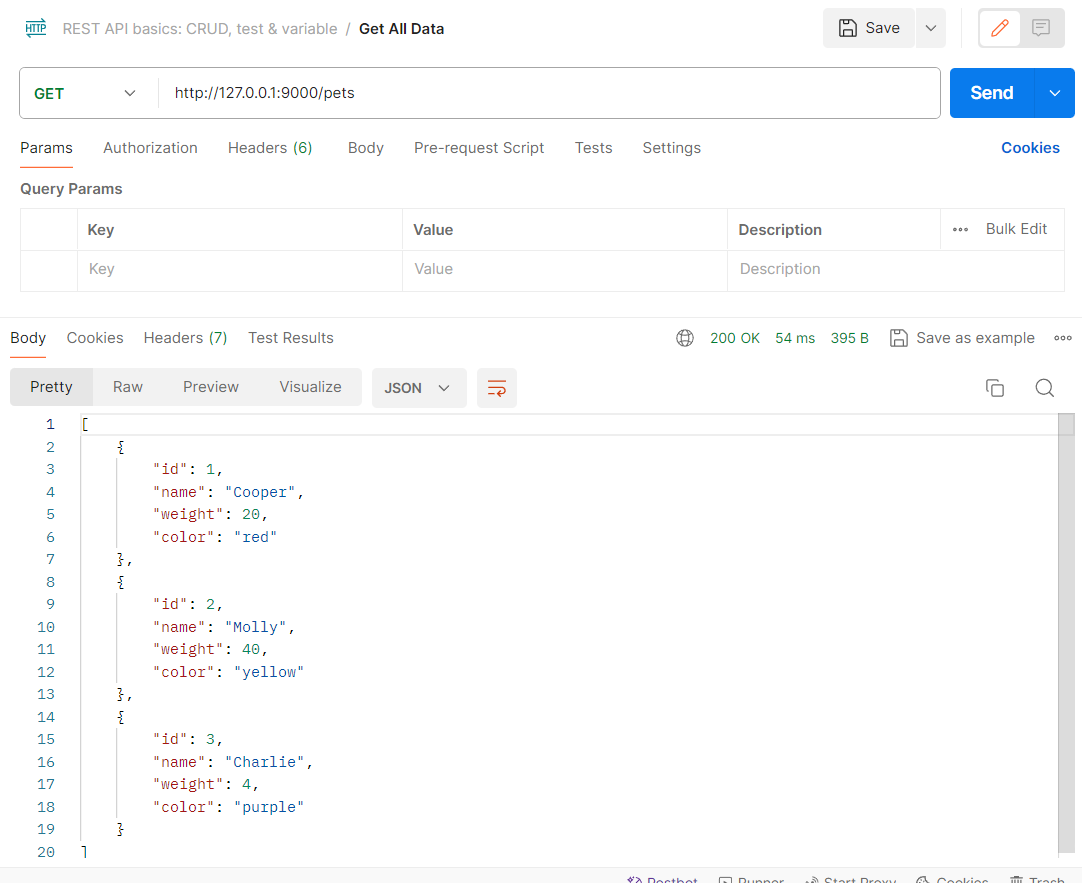
1. Hàm insert(pet)
2. Hàm update(id, changes)
3. Hàm getAll()
4. Hàm getById(id)
5. Hàm remove(id)

**Chương 3: Kết Quả Kiểm Thử**

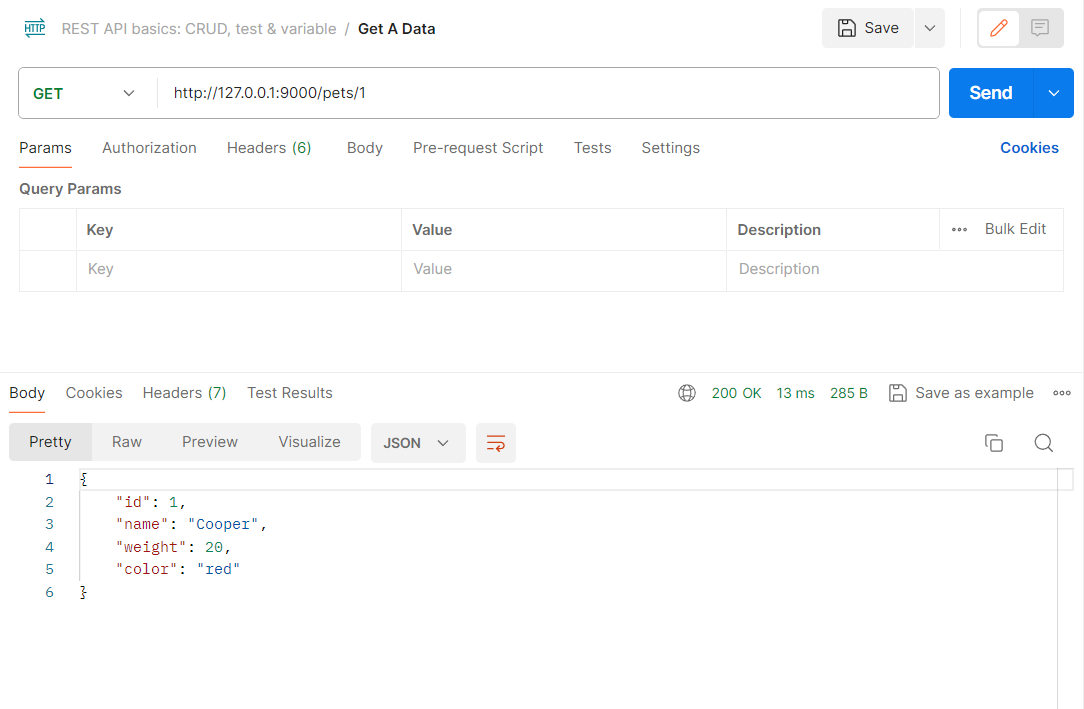
**[GET] “/”**

****

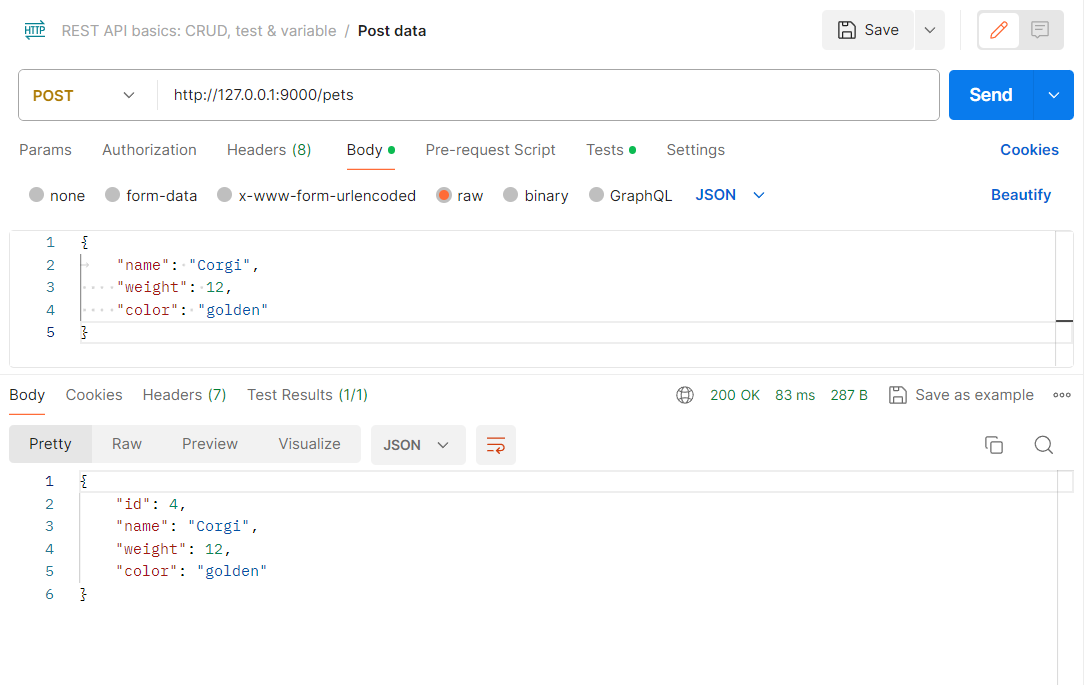
**[GET] “/pets”**

****

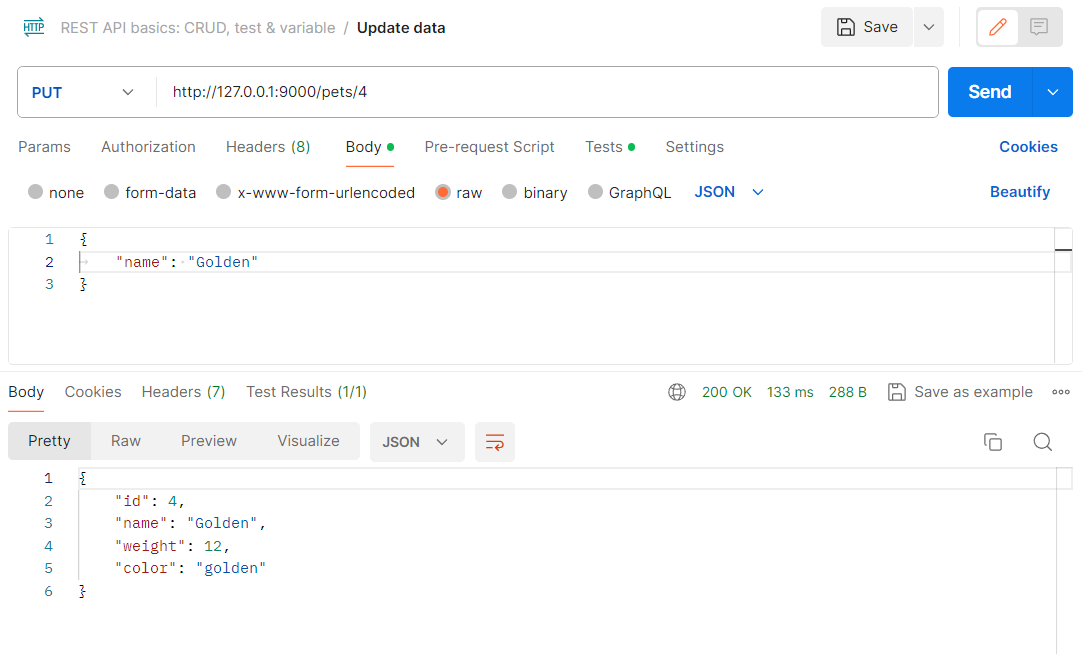
**[GET] “/pets/:id”**

****

**[POST] “/pets”**

****

**[PUT] “/pets/:id”**

****

**[DELETE] “/pets/:id”**

