

Bài thực hành – 03 – Routing

Mục tiêu:

- Thực hành xử lý Routing trên ExpressJS.
- Khởi tạo một project ExpressJS đơn giản xử lý Routing.

Bài 1: (4 điểm)

- Tạo một project ExpressJS quản lý thông tin sinh viên với yêu cầu cụ thể như sau:
 - o Xây dựng cấu trúc quản lý sinh viên theo cấu trúc json như hình bên cạnh.
 - o Trong đó:
 - sid – là mã số sinh viên.
 - name – là tên sinh viên.
 - scores – là danh sách điểm số từng môn của sinh viên
 - o Hãy viết các api thực hiện các yêu cầu như sau:
 - Liệt kê hết tất cả các sinh viên có trong danh sách.
 - Liệt kê đầy đủ thông tin của tất cả các sinh viên có trong danh sách và kèm theo tính toán điểm trung của tất cả các môn của sinh viên với công thức tính toán $GPA = (Math + Physics + Chemistry) / 3$
 - Trả về thông tin sinh viên có điểm GPA (điểm trung bình môn) cao nhất.
 - Trả về thông tin sinh viên có điểm GPA (điểm trung bình môn) thấp nhất.
 - Tìm kiếm sinh viên với SID biết trước.
 - Xóa thông tin của một sinh viên.
 - Cập nhật thông tin của môn sinh với SID biết trước
 - Thêm mới thông tin sinh viên mới.

```
[
  "Students": [
    {
      "sid": 1,
      "name": "Petter",
      "scores": [
        {
          "name": "math",
          "score": 8
        },
        {
          "name": "physics",
          "score": 8.5
        },
        {
          "name": "chemistry",
          "score": 9
        }
      ]
    },
    {
      "sid": 2,
      "name": "Mary",
      "scores": [
        {
          "name": "math",
          "score": 10
        },
        {
          "name": "physics",
          "score": 7.5
        },
        {
          "name": "chemistry",
          "score": 9.5
        }
      ]
    }
  ]
}
```

Bài 2: (4 điểm)

- Tạo một project Express thực hiện quản lý thông tin của danh sách sản phẩm trong file data.json có cấu trúc như hình bên.
- Yêu cầu xây dựng API để thực hiện các yêu cầu sau:
 - o Liệt kê tất cả các sản phẩm thuộc nhóm Food
 - o Liệt kê tất cả các sản phẩm thuộc nhóm Electric
 - o Thêm mới một sản phẩm vào nhóm Food
 - o Thêm mới một sản phẩm vào nhóm Electric
 - o Cập nhật thông tin của một sản phẩm thuộc nhóm Food.
 - o Cập nhật thông tin của một sản phẩm thuộc nhóm Electric.
 - o Xóa một sản phẩm trong nhóm Food.
 - o Xóa một sản phẩm trong nhóm Electric.
 - o Thêm một nhóm sản phẩm mới
 - o Xóa một nhóm sản phẩm (xóa luôn tất cả các sản phẩm có trong nhóm sản phẩm đó).
 - o Đổi tên của một nhóm sản phẩm hiện có.

```
[
  "categories": [
    {
      "foods": [
        {
          "name": "pork",
          "price": 100,
          "unit": "kilogram"
        },
        {
          "name": "beef",
          "price": 200,
          "unit": "kilogram"
        },
        {
          "name": "chicken",
          "price": 50,
          "unit": "once"
        }
      ]
    },
    {
      "electric": [
        {
          "name": "Tivi",
          "price": 1000
        },
        {
          "name": "refrigerator",
          "price": 1500
        },
        {
          "name": "air-conditioner",
          "price": 700
        }
      ]
    }
  ]
}
```

Bài 3: (2 điểm)

- Sử dụng lại cấu trúc file data.json của bài 2 ở trên, hãy viết chương trình ExpressJS để thực hiện các yêu cầu nâng cao như sau:
 - o Tự động tách các sản phẩm của nhóm Food ra thành file food.json.
 - o Tự động tách các sản phẩm của nhóm Electric ra thành file electric.json.
 - o Viết API phát sinh ngẫu nhiên 100 sản phẩm trong mỗi file food.json và electric.json.
 - o Gộp dữ liệu của 2 file food.json (gồm 100 sản phẩm) và file electric.json (gồm 100 sản phẩm) lại thành một file product.json (có hơn 200 sản phẩm) theo cấu trúc giống file data.json ở bài 2.
- Lưu ý:
 - o Mỗi chức năng/yêu cầu phải viết thành một API và API này được thực thi khi được call bằng ứng dụng Postman
 - o Thông tin các file đều được lưu trữ trên Server, và chỉ được truy cập và hiển thị dữ liệu thông qua API.