

Bài thực hành – 04 – Middleware

Mục tiêu:

- Thực hành xử lý Middleware trên ExpressJS.
- Khởi tạo một project ExpressJS với các Middleware để lọc (filter) và xử lý các request từ Client.

Bài 1: (4 điểm)

- Tạo một project ExpressJS để quản lý thông tin sản phẩm với định dạng thông tin sản phẩm như sau:

```
{ } data.json x
Users > ndhuy > Desktop > { } data.json > ...
1  {
2    "store": {
3      "brand": [
4        { "id": 1, "name": "Samsung" },
5        { "id": 2, "name": "Apple" },
6        { "id": 3, "name": "Lenovo" }
7      ],
8      "categories": [
9        {
10         "id": 1,
11         "name": "mobile",
12         "products": [
13           {
14             "id": 1, "name": "Galaxy S 21", "price": "1000", "brand_id": 1
15           },
16           {
17             "id": 2, "name": "iPhone 12 Pro Max", "price": "2000", "brand_id": 2
18           }
19         ]
20       },
21       {
22         "id": 2,
23         "name": "laptop",
24         "products": [
25           {
26             "id": 1, "name": "macBook Pro M1", "price": 2000, "brand_id": 2
27           },
28           {
29             "id": 2, "name": "X1 Carbon", "price": 2500, "brand_id": 3
30           },
31           {
32             "id": 3, "name": "Galaxy Book Pro", "price": 3100, "brand_id": 1
33           }
34         ]
35       }
36     ]
37   }
38 }
```

- Xây dựng các API để thực hiện các yêu cầu sau:
 - o Xây dựng một Middleware để ghi log lại thông tin của Request như request time, request, method, request URL của mỗi request gửi đến server vào màn hình terminal của server.
 - o Xây dựng Middleware để resetup phần Response Header cho các response xử lý phía sau là
 - **Status Code** = 200 – thông báo xử lý request thành công
 - **Content-Type** = JSON – định dạng dữ liệu response về luôn là kiểu JSON
 - o Xây dựng các API xử lý sau:
 - **GET /brands** → trả về danh sách Brand
 - **GET /brands/id/:id** → trả về thông tin của một Brand có Brand ID tương ứng
 - **GET /categories** → trả về danh sách Categories
 - **GET /categories/id/:id** → trả về thông tin của một
 - **GET /products** → trả về danh sách tất cả các Product
 - *Lưu ý cần phải bổ sung thêm thông tin Brand Name và Category Name của mỗi Product khi response về cho Client.*
 - **GET /products/id/:id** → trả về thông tin của một Product có product ID tương ứng.
 - **GET /brands/id/:id/products** → trả về tất cả các Product thuộc Brand với Brand ID tương ứng
 - **GET /categories/id/:id/products** → trả về tất cả các Product thuộc Categories với Category ID tương ứng
 - **POST /product** → tạo thông tin product mới với
 - *Sử dụng Raw để tạo thông tin product dạng JSON trong request*
 - *Dựa vào thông tin Category trong JSON object ở trên, cần phải xử lý push thông tin của Product nào vào Category có ID tương ứng.*
 - **PUT /product** → cập nhật thông tin của product (dựa vào kỹ thuật của câu trên để cập nhật thông tin của product tương ứng)
 - **DELETE /product/id/:id** → xóa thông tin của một Product có ID tương ứng

Bài 2: (4 điểm)

- Xây dựng ứng dụng ExpressJS quản lý thông tin người dùng.
 - o Thông tin người dùng sẽ được lưu trữ trong file data.json có cấu trúc giống hình bên dưới.
 - o Trong file đặc tả bên dưới thì một user sẽ có 4 thông tin là: user_id, username, password, full_name.
 - o Một user sẽ thuộc về một role_group, hiện có 3 role_group:
 - user – là những người dùng bình thường.
 - staff – là những người dùng thuộc nhóm nhân viên quản lý quy trình bán hàng trong hệ thống.
 - admin – là những người dùng có quyền quản trị hệ thống.
 - o Thông tin current_user_id để cho biết ID của user cuối cùng hiện đang bằng bao nhiêu, điều này sẽ hỗ trợ cho việc tạo user_id mới.
 - Hãy xây dựng các API để thực hiện các chức năng sau:
-

- Liệt kê tất cả danh sách user đang có trong hệ thống
- Liệt kê tất cả nhân viên thuộc nhóm user.
- Liệt kê tất cả các nhân viên thuộc nhóm staff.
- Liệt kê tất cả các nhân viên thuộc nhóm admin.
- Hiển thị thông tin của một nhân viên với username và password được gửi kèm theo dạng RAW trong Request package.
- Thêm một nhân viên với đối với user thuộc quyền admin (cần tuyến thêm username và password kèm theo request package để kiểm tra quyền ở phía API)
- Thực hiện chuyển role cho một user với user_id được truyền vào.
- Thực hiện cập nhật password cho một user với user_id được truyền vào.
- Thực hiện cập nhật full_name cho một user với user_id được truyền vào.
- Thực hiện cập nhật cả password và full_name của một user với user_id được truyền vào.
- Xóa một user ra khỏi danh sách với user_id được truyền vào.

```

data.json x
Users > ndhuy > Desktop > data.json > ...
1  {
2    "role_group": {
3      "user": {
4        "role_id": 1,
5        "users": [
6          { "user_id": 1, "username": "user_01", "password": "123456", "full_name": "Peter" },
7          { "user_id": 2, "username": "user_02", "password": "654321", "full_name": "Mary" },
8          { "user_id": 3, "username": "user_03", "password": "112233", "full_name": "Henry" },
9          { "user_id": 4, "username": "user_04", "password": "332211", "full_name": "Mac" }
10       ]
11     },
12     "staff": {
13       "role_id": 2,
14       "users": [
15         { "user_id": 5, "username": "user_05", "password": "999888", "full_name": "Julie" }
16       ]
17     },
18     "admin": {
19       "role_id": 3,
20       "users": [
21         { "user_id": 6, "username": "admin_06", "password": "456789", "full_name": "Boss" },
22         { "user_id": 7, "username": "admin_07", "password": "778899", "full_name": "Bill" }
23       ]
24     }
25   },
26   "config": {
27     "current_user_id": 7
28   }
29 }

```

Bài 3: (2 điểm)

- Sử dụng 2 file data.json ở trên để thực hiện một hệ thống quản lý thông tin hàng hóa như sau:
 - Thực hiện các công việc đối với user thuộc role_group là user:
 - Liệt kê thông tin tất cả các sản phẩm.
 - Liệt kê thông tin của các sản phẩm theo Brand với Brand Name được truyền vào.
 - Liệt kê thông tin của các sản phẩm theo Category với Category Name được truyền vào.
 - Hiển thị thông tin chi tiết của một Product với Product ID được truyền vào.
 - Thực hiện các công việc đối với user thuộc role_group là staff:
 - Thêm Brand mới.
 - Thêm Category mới.
 - Thêm một Product mới.
 - Cập nhật tên của một Brand.
 - Cập nhật tên của một Category.
 - Cập nhật thông tin của một Product.
 - Xóa một Brand.
 - Xóa một Category.
 - Xóa một Product.
 - Thực hiện các công việc đối với user thuộc role_group là Admin:
 - Thay đổi role cho một user.
 - Cập nhật thông tin của user.
 - Thêm một user mới.
 - Xóa một user.
 - Liệt kê danh sách tất cả các user.
 - Hiển thị thông tin của một user với user_id truyền vào.
 - Lưu ý:
 - Các request được truyền về bắt buộc phải truyền theo username và password.
 - Cần phải xây dựng một middleware để kiểm tra quyền của các request để xem có thuộc role để thực hiện các chức năng quy định theo role hay không?
 - Nếu thỏa thì next sang route xử lý chính.
 - Nếu không thỏa thì trả về lỗi 403 – Access Denied.
-