**Nhóm 14 – NTO23: EXPRESS JS**

Phản hồi thắc mắc Seminar.

**Câu 1: Làm thế nào để thiết kế một APT RESTFul đơn giản dễ bảo trì bằng Express JS ?**

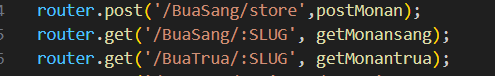
* Việc thiết kế restful api bằng express cơ bản cũng giống như các framework khác, người dùng cần nắm vững javascrips cũng như cấu trúc thiết kế api của nodejs thì có thể thiết kế được 1 restful api bằng expres js. Bên cạnh đó chúng ta cần hiểu được luồng dữ liệu cũng như các routes của các api thì việc bảo trì cũng sẽ trở nên vô cùng đơn giản.



Ví dụ: API verify email của user.

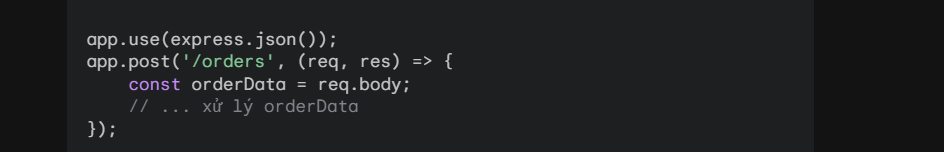
**Câu 2: Cách để truyền tham số và xử lí tuyến đường trong Express JS.**

* Về vấn đề truyền tham số, trong Express lập trình viên có thể truyền thông qua việc cấu trúc sẵn trên hệ thống URL:



 Sử dụng middleware express.json() hoặc express.urlencoded() để phân tích cú pháp dữ liệu JSON hoặc form-data.

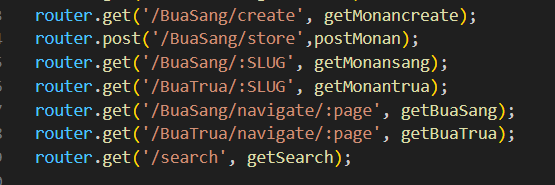
 Truy cập giá trị tham số thông qua req.body trong controller.



-Routing:

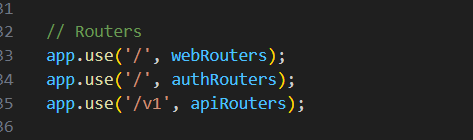
 Sử dụng các phương thức HTTP (GET, POST, PUT, DELETE, ...) để xác định hành động cho từng tuyến đường.

 Sử dụng app.METHOD(PATH, HANDLER) để liên kết tuyến đường với hàm xử lý



 Sử dụng express.Router() để tạo các module định tuyến riêng biệt, giúp tổ chức mã nguồn tốt hơn.

 Đăng ký router vào ứng dụng Express bằng app.use(PATH, ROUTER).



**Câu 3: Có những gì để hỗ trợ phân quyền.**

* Có thể xây dựng các Middleware phân quyền :

 Bạn có thể tự xây dựng middleware để kiểm tra quyền truy cập dựa trên vai trò người dùng, token JWT hoặc các cơ chế xác thực khác.

 Đây là cách linh hoạt nhất nhưng đòi hỏi bạn phải tự xử lý logic phân quyền.



* Thư viện phân quyền : ACL, RBAC ,OAUTH 2.0

**Câu 4: Phương pháp nào tốt nhất để bảo mật trong Express JS?**

Bảo mật là một yếu tố cực kỳ quan trọng trong việc phát triển ứng dụng web, đặc biệt là khi sử dụng Express.js. Không có phương pháp nào được coi là tốt nhất vì nó còn tùy vào các yếu tố như ngân sách và thời gian, kỹ năng của đội ngũ phát triển, loại ứng dụng và môi trường. Dưới đây là một số phương pháp và thực hành để bảo mật ứng dụng Express.js của bạn.

1. Bảo vệ ứng dụng khỏi các tấn công phổ biến:

* Sử dụng helmet, xác thực đầu vào, sử dụng bcrypt để băm mật khẩu, ngăn chặn Cross-Site Request Forgery (CSRF) ,Sử dụng Passport.js, phân quyền dựa trên vai trò (RBAC), JSON Web Tokens (JWT):

3. Bảo mật dữ liệu:

* Mã hóa dữ liệu nhạy cảm
* Sử dụng HTTPS
* Cập nhật thường xuyêm

4. Kiểm tra và giám sát bảo mật:

* Sử dụng công cụ quét lỗ hổng bảo mật
* Giám sát log.
* Thực hiện kiểm tra thâm nhập

**Câu 5: Express có phù hợp với các dự án có ngân sách nhỏ hay không?**

* Express js rất phù hợp với các dự án có ngân sách nhỏ vì dễ bảo quản và sử dụng cũng như dễ dàng mở rộng và phát triển.