HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÁO CÁO

THỰC TẬP CƠ SỞ

Giảng viên: Kim Ngọc Bách

Tuần 9: 10/5/2025

Sinh viên: Nguyễn Khắc Trường B22DCCN884

MỤC LỤC

A. Kiến thức học được ở tuần này

Giới thiệu

Tiếp tục với kiến thức đã học được ở tuần trước, tuần này em tiếp tục tìm hiểu về các thư viện mà em định dùng trong phần backend của dự án.

Phần 1: Tự động hóa getter/setter và tối ưu code với Lombok

Trong các mô hình lập trình hướng đối tượng, các phương thức getter/setter, constructor hoặc equals/hashCode thường được viết lặp đi lặp lại. Thư viện lombok giúp giảm đáng kể sự lặp lại này bằng cách sử dụng annotation.

1. Lombok hỗ trợ những gì?

- @Getter, @Setter: Tự động tạo getter/setter cho các field.
- @AllArgsConstructor, @NoArgsConstructor: Sinh constructor với hoặc không có tham số.
- @Builder: Tạo class với pattern builder một cách dễ dàng.
- @Data: Kết hợp nhiều annotation hữu ích như getter, setter, toString, equals, hashCode.

2. Cách hoạt động

- Trong lúc biên dịch (compile time), Lombok sẽ sinh mã Java tương ứng dựa trên các annotation.
- Developer không cần tự viết getter/setter, giúp code ngắn gọn và dễ đọc.
- Cần bật hỗ trợ annotation processing trong IDE (như IntelliJ hoặc Eclipse).

3. Tác dụng

- Tiết kiệm thời gian lập trình.
- Giảm boilerplate code.
- Tăng khả năng đọc hiểu và bảo trì mã nguồn.

Phần 2: Xác thực người dùng với JWT và jjwt

JSON Web Token (JWT) là phương pháp xác thực người dùng phổ biến trong ứng dụng web và API. Bộ thư viện jjwt (gồm jjwt-api, jjwt-impl, jjwt-jackson) giúp tạo và xác thực JWT đơn giản trong Spring Security.

1. JWT bao gồm những gì?

- Header: Thông tin thuật toán ký (HS256, RS256,...)
- Payload: Chứa thông tin người dùng (username, role,...)
- Signature: Chữ ký dùng để xác thực tính toàn vẹn token.

2. Cách hoạt động

- Khi người dùng đăng nhập thành công, server sẽ tạo JWT với thông tin người dùng và gửi về client.
- Mỗi request tiếp theo từ client sẽ kèm JWT trong header Authorization.
- Server dùng thư viện jjwt để xác thực token, giải mã payload và xác định quyền truy cập.

3. Tác dụng

- Stateless: Không cần lưu phiên (session) trên server.
- Bảo mật và hiệu quả: Token được ký và không thể sửa đổi.
- Dễ dàng tích hợp với Spring Security bằng filter hoặc config tùy chỉnh.

Phần 3: Mapping đối tượng với ModelMapper

Việc chuyển đổi giữa entity và DTO là cần thiết để tách biệt dữ liệu hiển thị và lưu trữ. ModelMapper là thư viện giúp ánh xạ tự động giữa các class có cấu trúc tương đồng.

1. Tính năng chính

- Tự động ánh xạ field cùng tên giữa hai class.
- Cho phép tùy chỉnh ánh xạ (custom mapping).
- Hỗ trợ ánh xạ lồng nhau (nested object mapping).

2. Cách hoạt động

- Khởi tạo một đối tượng ModelMapper.
- Gọi modelMapper.map(source, Destination.class) để ánh xạ dữ liệu.
- Có thể cấu hình để bỏ qua hoặc ánh xạ riêng một số field đặc biệt.

3. Tác dụng

- Giảm thời gian viết code chuyển đổi dữ liệu giữa DTO và Entity.
- Dễ bảo trì khi có thay đổi về cấu trúc dữ liệu.
- Giúp API trả về dữ liệu gọn gàng, đúng định dạng mong muốn.

Phần 4: Upload ảnh với Cloudinary

Cloudinary là một dịch vụ lưu trữ và xử lý hình ảnh/video trên cloud. Thư viện cloudinary-http44 cho phép tích hợp Cloudinary vào ứng dụng Java.

1. Chức năng chính

- Upload ảnh lên cloud.
- Resize, crop, chuyển định dạng ảnh tự động.
- Lưu URL ảnh trả về để dùng trong ứng dụng.

2. Cách hoạt động

- Cấu hình tài khoản Cloudinary với cloud_name, api_key, api_secret.
- Gọi API upload từ Java client để đẩy ảnh lên.
- Nhận lại URL ảnh từ Cloudinary và lưu vào database hoặc trả về frontend.

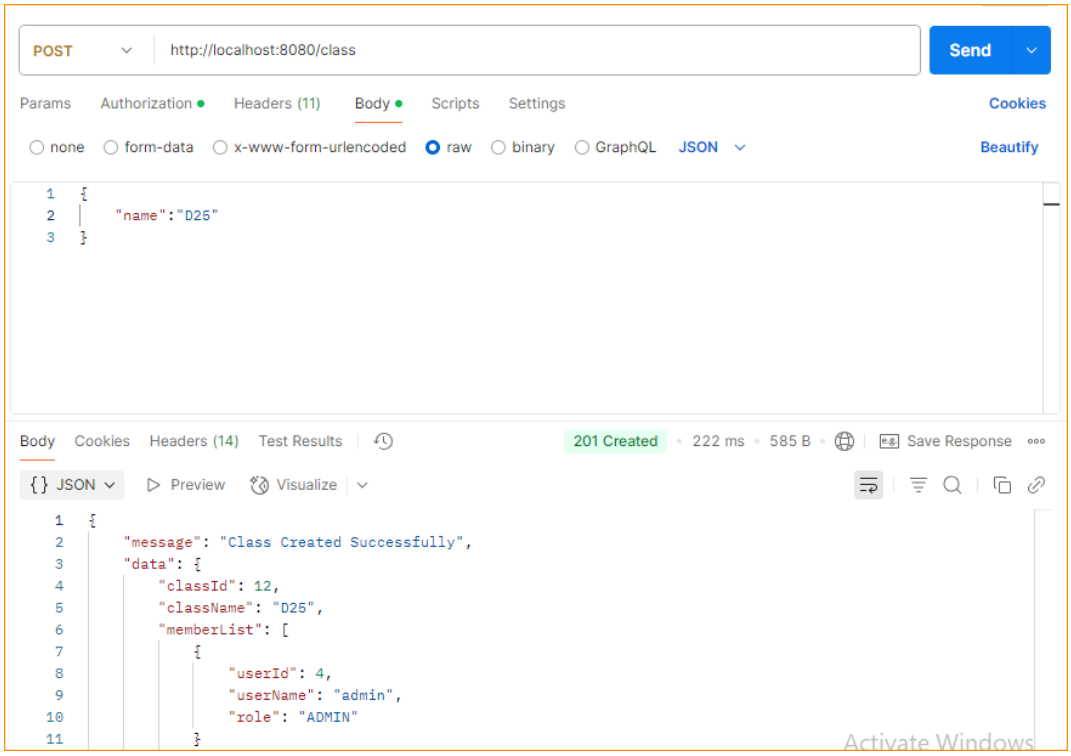
3. Tác dụng

- Không cần tự xây dựng server lưu trữ ảnh.
- Tối ưu tốc độ tải ảnh cho người dùng.
- Hỗ trợ CDN và các tính năng nâng cao như watermark, lazy load,…

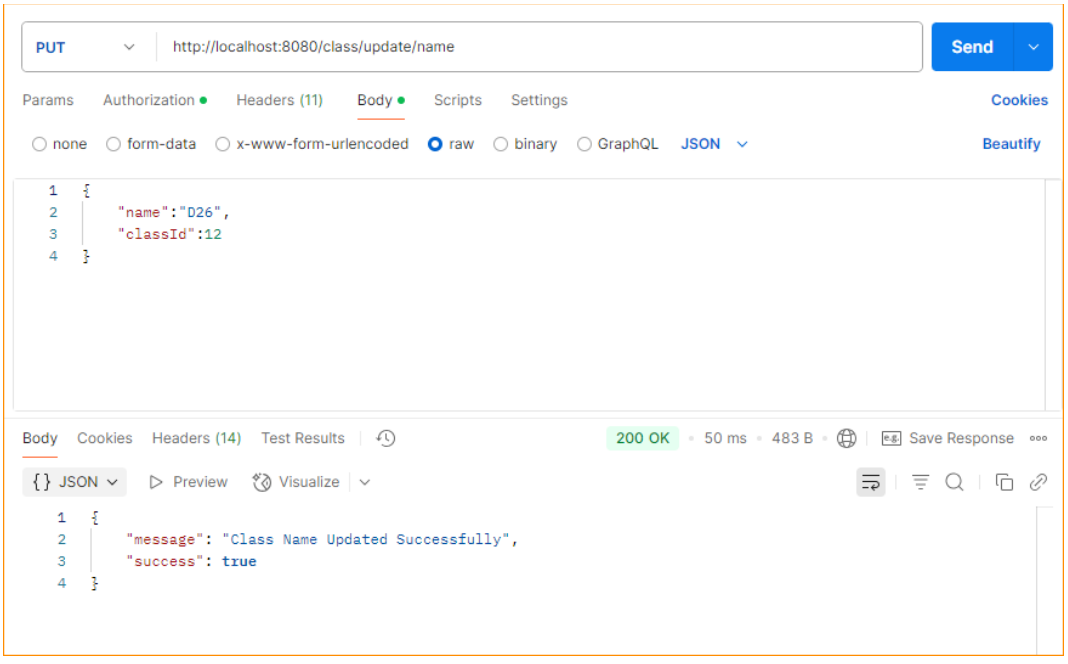
B. Tiến độ dự án

- Hoàn thiện các API liên quan đến lớp học (chưa bao gồm api liên quan set[tập từ] trong lớp học):

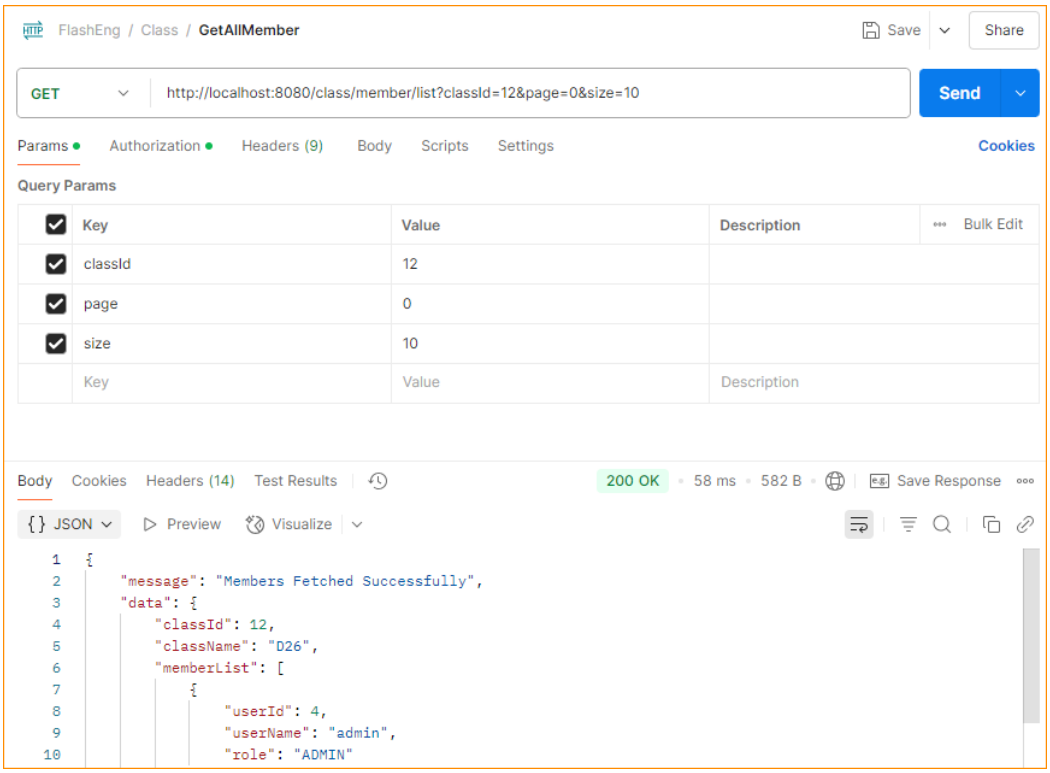
+ Tạo lớp: Người dùng nhập tên lớp để tạo lớp.



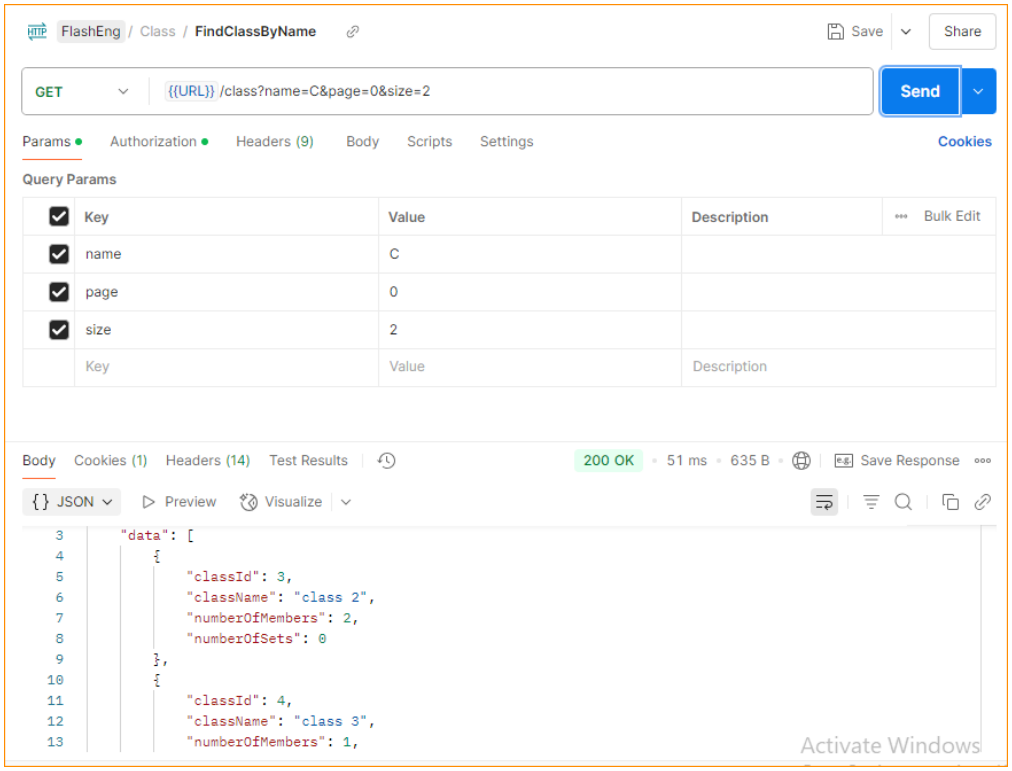
+ Sửa tên lớp: Người dùng nhập tên lớp và Id của lớp để sửa tên.



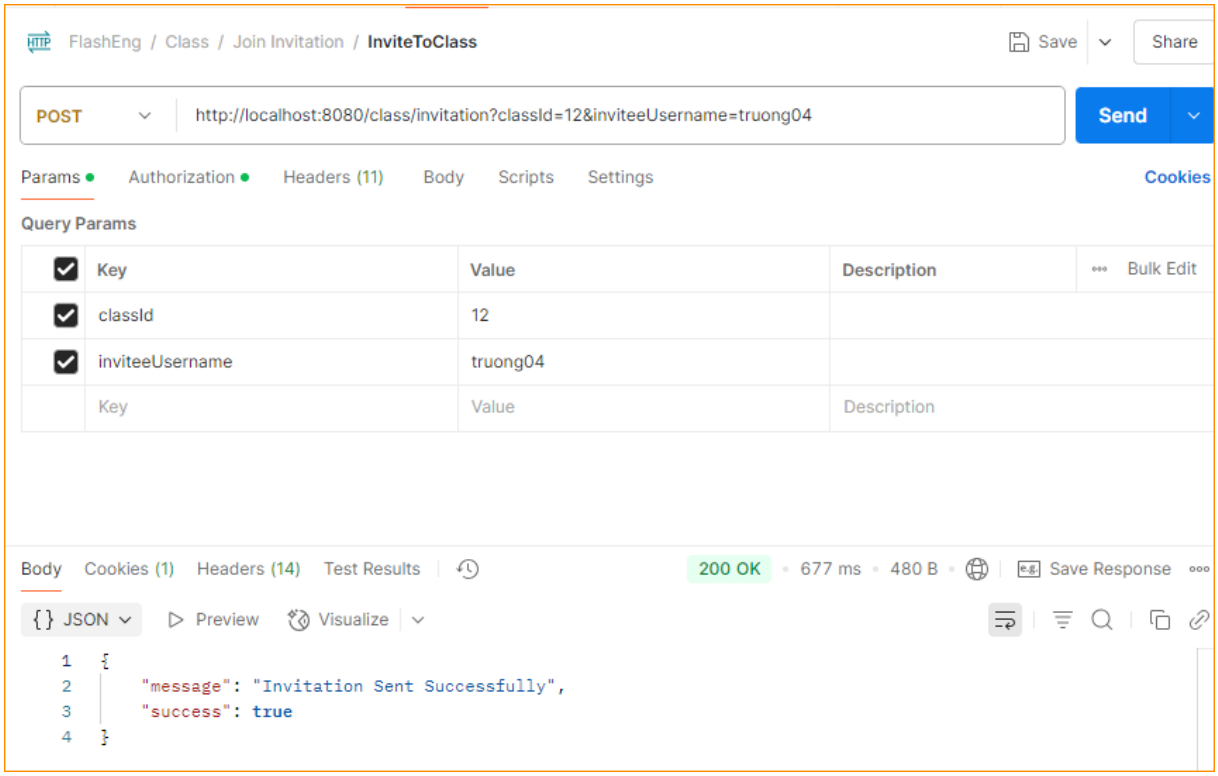
+ Lấy danh sách thành viên lớp: Người dùng nhập id lớp và page,size giúp cho phân trang ở phía FE và token giúp xác thực xem có phải thành viên nhóm không.



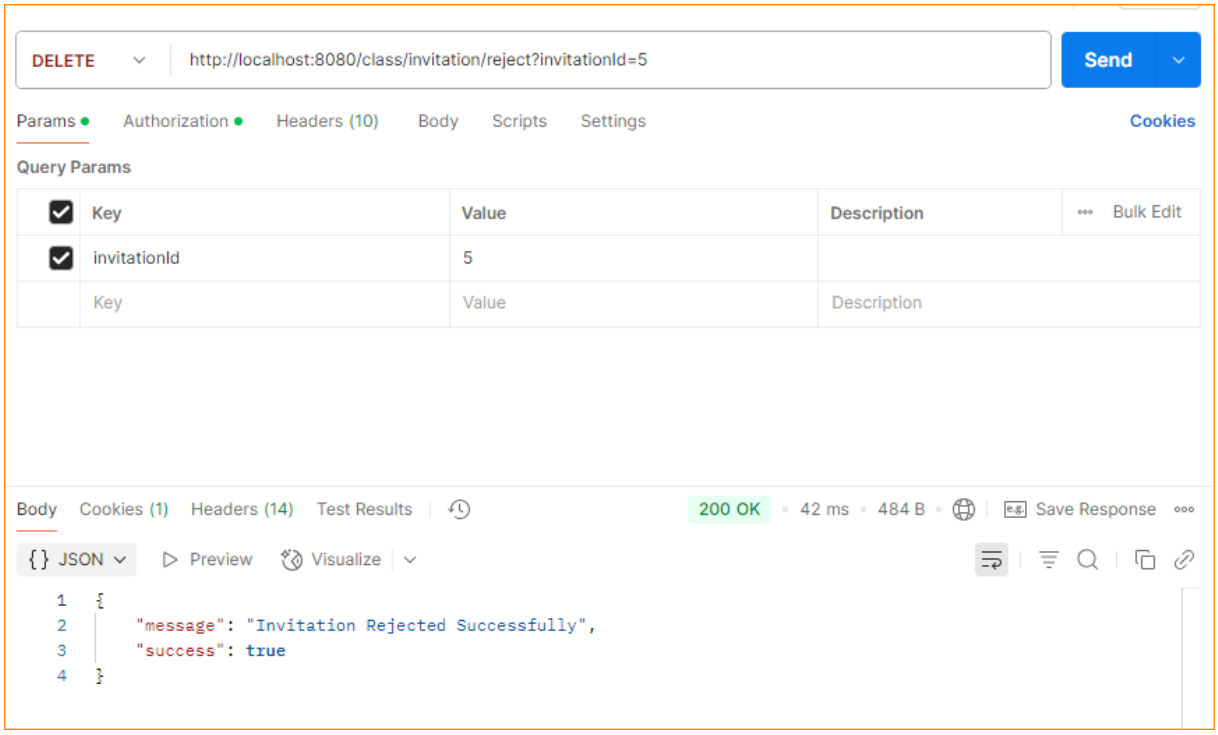
+ Tìm lớp: Người dùng nhập tên lớp cần tìm cùng với page, size giúp phân trang trên FE



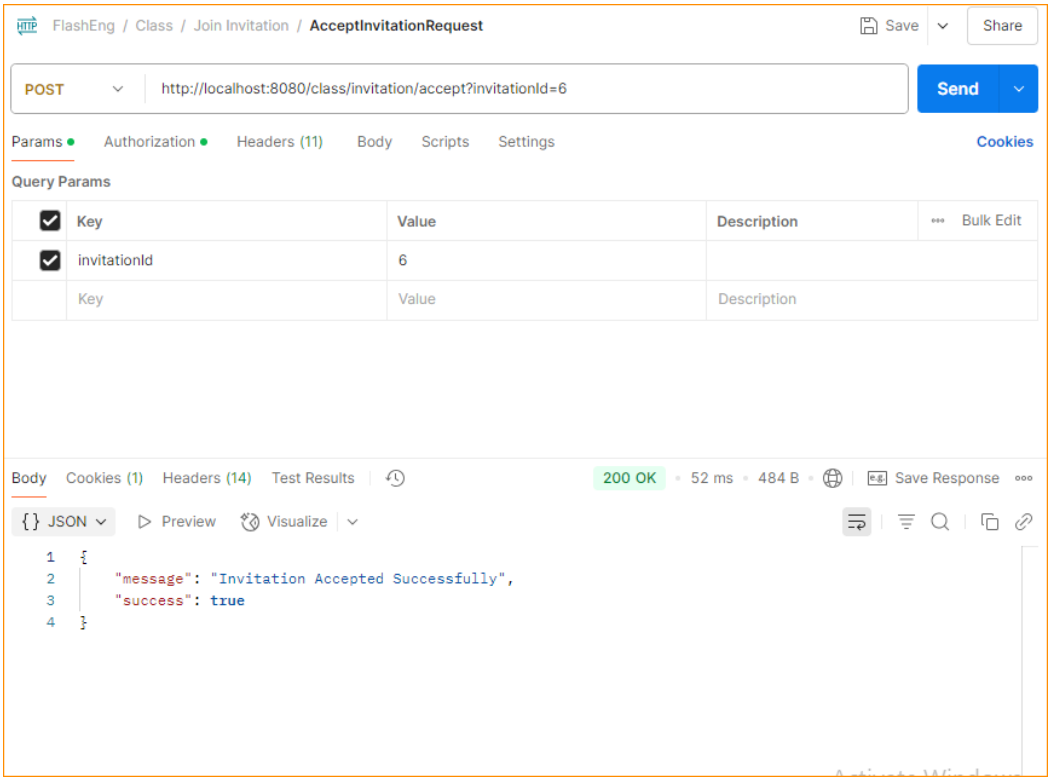
+ Mời vào lớp: Người dùng nhập id lớp học cùng với username của người muốn mời vào nhóm (Tương tự với xin gia nhập nhóm nhập classId và token).



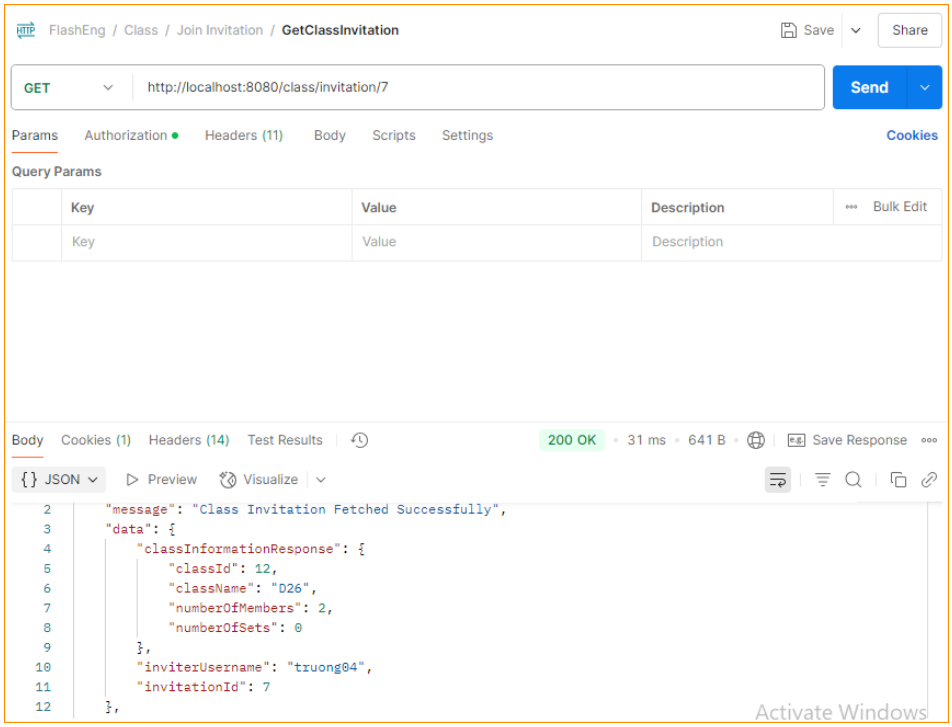
+ Từ chối lời mời gia nhập: Người dùng nhập invitationId (Tương tự với xin gia nhập nhóm nhập join_request_id).



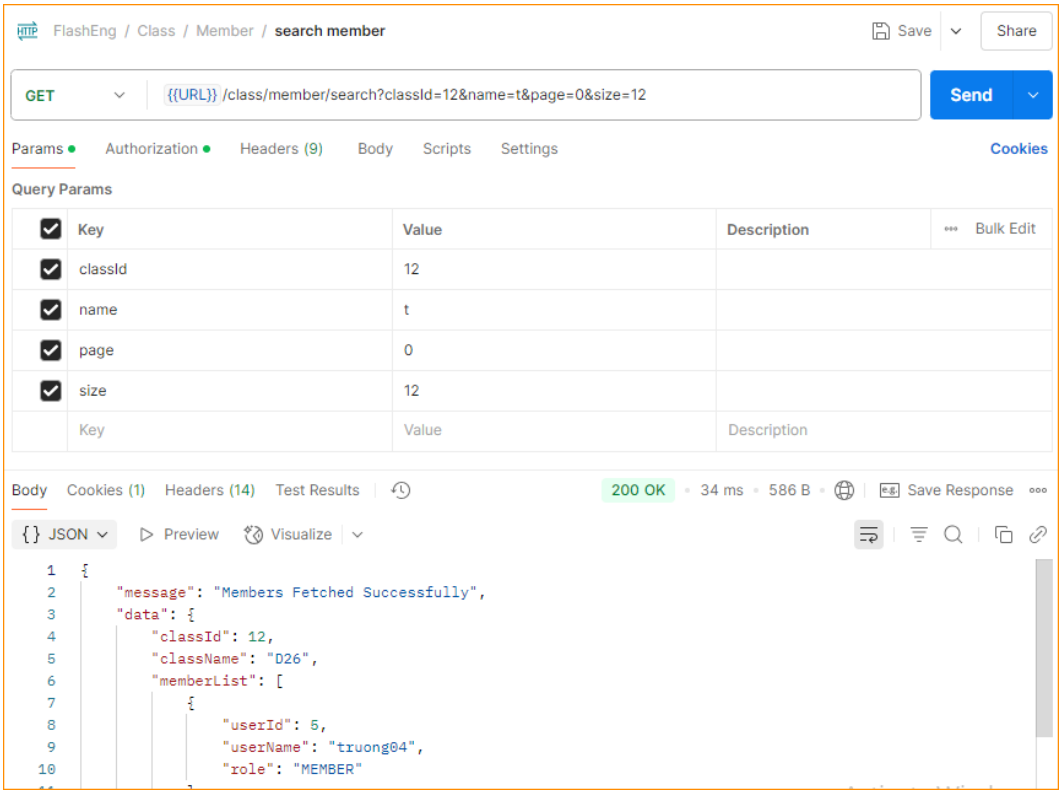
+ Đồng ý lời mời gia nhập: Người dùng nhập invitationId (Tương tự với xin gia nhập nhóm nhập join_request_id).



+ Lấy thông tin lời mời: Người dùng nhập invitationId (Tương tự với xin gia nhập nhóm nhập join_request_id).



+ Tìm member trong nhóm: Người dùng nhập classId, username của người cần tìm, page và size giúp cho việc phân trang bên FE.



+ Rời nhóm: Người dùng nhập classId để chọn lớp cần rời và token giúp xác minh danh tính người dùng.