

Hình ảnh mô tả quy trình xác thực và ủy quyền người dùng bằng JWT (JSON Web Token) trong một ứng dụng Client-Server. Quá trình này bao gồm ba bước chính:

**1. User Registration (Đăng ký người dùng)**

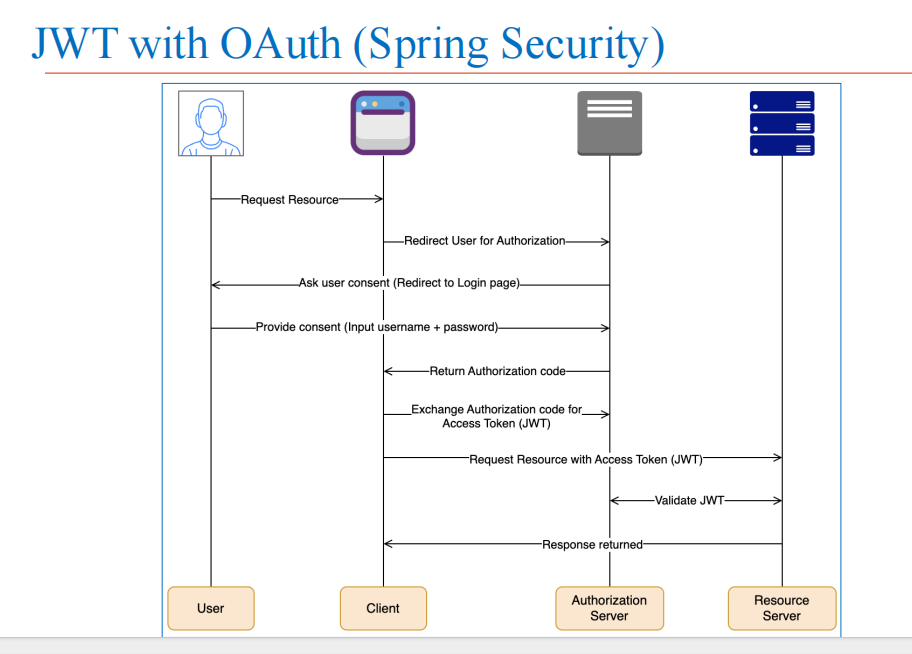
* **Bước 1**: Client gửi yêu cầu POST /api/auth/signup với thông tin { username, email, role, password }.
* **Bước 2**: Server kiểm tra xem người dùng đã tồn tại chưa, nếu chưa thì lưu vào cơ sở dữ liệu.
* **Bước 3**: Server phản hồi về client một thông báo **"Registered successfully!"**.

**2. User Login (Đăng nhập người dùng)**

* **Bước 1**: Client gửi yêu cầu POST /api/auth/signin với { username, password }.
* **Bước 2**: Server xác thực thông tin đăng nhập và nếu hợp lệ, tạo một **JWT token** bằng cách mã hóa thông tin người dùng với một khóa bí mật.
* **Bước 3**: Server trả về phản hồi { token, type, user info, authorities }.

**3. Access Resource (Truy cập tài nguyên)**

* **Bước 4**: Client gửi yêu cầu truy cập tài nguyên bằng cách đính kèm **JWT** vào Authorization Header.
* **Bước 5**: Server kiểm tra chữ ký JWT, xác thực thông tin người dùng, kiểm tra quyền hạn của họ.
* **Bước 6**: Server phản hồi với dữ liệu dựa trên quyền của người dùng.



Hình này mô tả quy trình **xác thực và ủy quyền người dùng** bằng **JWT với OAuth2 trong Spring Security**. Quy trình này giúp người dùng truy cập tài nguyên một cách bảo mật bằng **OAuth2 Authorization Code Flow** và sử dụng **JWT** để xác thực.

**Các thành phần chính:**

1. **User (Người dùng)**: Người dùng yêu cầu truy cập tài nguyên.
2. **Client (Ứng dụng hoặc trình duyệt)**: Trung gian gửi yêu cầu xác thực.
3. **Authorization Server (Máy chủ xác thực)**: Xác thực người dùng và cấp **Authorization Code** & **JWT Token**.
4. **Resource Server (Máy chủ tài nguyên)**: Lưu trữ tài nguyên cần được bảo vệ.

**Quy trình xác thực và ủy quyền:**

1. **User yêu cầu tài nguyên**
   * Người dùng gửi yêu cầu truy cập tài nguyên thông qua Client.
2. **Client chuyển hướng người dùng đến Authorization Server**
   * Authorization Server yêu cầu người dùng đăng nhập hoặc cung cấp quyền truy cập.
3. **Người dùng nhập thông tin đăng nhập**
   * Người dùng nhập username và password, sau đó cung cấp sự đồng ý.
4. **Authorization Server cấp Authorization Code**
   * Nếu xác thực thành công, Authorization Server gửi một **Authorization Code** về Client.
5. **Client đổi Authorization Code lấy Access Token (JWT)**
   * Client gửi Authorization Code lên Authorization Server để đổi lấy **Access Token (JWT)**.
6. **Client dùng JWT để truy cập tài nguyên**
   * Client gửi yêu cầu đến Resource Server, đính kèm **JWT Token**.
7. **Resource Server xác thực JWT**
   * Resource Server kiểm tra chữ ký JWT và xác thực quyền truy cập.
8. **Resource Server phản hồi dữ liệu**
   * Nếu hợp lệ, Resource Server trả về dữ liệu mà người dùng yêu cầu.

**Tóm lại:**

* **OAuth2** giúp cấp quyền truy cập an toàn mà không cần chia sẻ mật khẩu.
* **JWT** giúp xác thực nhanh chóng mà không phải truy vấn cơ sở dữ liệu nhiều lần.
* **Spring Security** kết hợp OAuth2 + JWT để bảo vệ API một cách bảo mật và hiệu quả.