**Quy trình xác thực người dùng bằng JWT trong REST API**

**JWT (JSON Web Token) là phương pháp phổ biến để xác thực và phân quyền trong các hệ thống RESTful API. Dưới đây là mô tả chi tiết về quy trình xác thực người dùng bằng JWT và vòng đời của token.**

**1. Sơ đồ tổng quát quy trình xác thực bằng JWT**

**[Client] --(1. Gửi thông tin đăng nhập)--> [Server]**

**[Server] --(2. Xác thực + sinh JWT)--> [Client]**

**[Client] --(3. Gửi JWT qua request)--> [Server]**

**[Server] --(4. Kiểm tra JWT)--> [Cho phép hoặc từ chối truy cập tài nguyên]**

**2. Chi tiết các bước xác thực JWT**

**✅ Bước 1: Client gửi thông tin đăng nhập**

* **Người dùng nhập username/password vào form.**
* **Trình duyệt gửi HTTP request (thường là POST) tới endpoint /login hoặc /auth/login.**

**✅ Bước 2: Server xác thực và sinh token**

* **Server kiểm tra thông tin đăng nhập hợp lệ hay không.**
* **Nếu đúng:**
  + **Tạo JWT chứa các thông tin như userId, role, iat, exp, v.v.**
  + **Ký token bằng secret key hoặc private key.**
* **Gửi JWT về client qua response (thường đặt trong body hoặc cookie).**

**✅ Bước 3: Client lưu trữ token**

* **Client nhận token và lưu vào:**
  + **localStorage hoặc sessionStorage (trong trình duyệt).**
  + **Hoặc cookie nếu muốn gửi kèm tự động trong mỗi request.**

**✅ Bước 4: Client gửi request kèm JWT**

* **Mỗi khi gọi các API cần xác thực, client gửi JWT trong header:**

**Authorization: Bearer <jwt\_token>**

**✅ Bước 5: Server xác minh token**

* **Server kiểm tra:**
  + **Token có hợp lệ không? (giải mã, kiểm tra chữ ký)**
  + **Token đã hết hạn chưa? (exp)**
* **Nếu hợp lệ → xử lý request và trả dữ liệu.**
* **Nếu không → trả lỗi 401 Unauthorized.**

**3. Vòng đời của JWT**

1. **Phát hành: Khi người dùng đăng nhập thành công.**
2. **Lưu trữ: Client lưu token.**
3. **Sử dụng: Gửi kèm theo mỗi request để truy cập tài nguyên cần xác thực.**
4. **Hết hạn: Token sẽ không hợp lệ sau thời gian exp, cần login lại hoặc dùng refresh token nếu có.**

**4. Ghi chú bảo mật**

* **Luôn sử dụng HTTPS để tránh lộ token.**
* **Đặt thời gian sống (exp) ngắn hợp lý.**
* **Không lưu token trong localStorage nếu có nguy cơ XSS.**
* **Cân nhắc dùng Refresh Token để tăng bảo mật.**

**✅ Kết luận: JWT là công cụ mạnh mẽ để xác thực người dùng trong hệ thống REST API. Hiểu rõ quy trình và vòng đời của token sẽ giúp bạn xây dựng hệ thống bảo mật và hiệu quả.**