**Spring framework**

# **7. OAuth2**

# **1. Tìm hiểu về khái niệm và cách thức hoạt động của OAuth2**

**OAuth2 (Open Authorization 2.0)** là một giao thức ủy quyền mở, cho phép các ứng dụng bên thứ ba truy cập vào tài nguyên của người dùng trên một dịch vụ mà không cần phải tiết lộ mật khẩu của người dùng. Nó được sử dụng rộng rãi để cho phép truy cập vào các API một cách bảo mật mà không yêu cầu người dùng trực tiếp cung cấp thông tin xác thực cho các ứng dụng.

### **Các thành phần chính của OAuth2:**

1. **Resource Owner**: Chủ sở hữu tài nguyên, thường là người dùng. Họ có quyền kiểm soát tài nguyên của mình và có thể cấp quyền truy cập cho các ứng dụng bên thứ ba.
2. **Client**: Ứng dụng bên thứ ba muốn truy cập tài nguyên của người dùng. Client cần phải được ủy quyền bởi người dùng để truy cập tài nguyên.
3. **Resource Server**: Máy chủ lưu trữ các tài nguyên mà client muốn truy cập, chẳng hạn như API của Google, Facebook, v.v.
4. **Authorization Server**: Máy chủ chịu trách nhiệm xác thực người dùng và cấp quyền truy cập thông qua **Access Token**.
5. **Access Token**: Một mã thông báo được cấp bởi Authorization Server cho phép client truy cập tài nguyên trên Resource Server thay mặt cho người dùng.

### **Quy trình hoạt động của OAuth2:**

OAuth2 có nhiều loại luồng (flows) tùy thuộc vào hoàn cảnh và loại ứng dụng. Trong đó, **Authorization Code Flow** là một trong những luồng phổ biến nhất và thường được sử dụng cho các ứng dụng web. Dưới đây là cách hoạt động của **Authorization Code Flow**:

1. **Bước 1: Người dùng bắt đầu quá trình ủy quyền**
   * Người dùng truy cập ứng dụng bên thứ ba (client) và yêu cầu ứng dụng đó thực hiện một hành động yêu cầu quyền truy cập tài nguyên từ một dịch vụ (ví dụ, API của Google).
2. **Bước 2: Client chuyển hướng người dùng đến Authorization Server**
   * Ứng dụng bên thứ ba (client) sẽ chuyển hướng người dùng đến **Authorization Server** của dịch vụ (ví dụ: Google) để yêu cầu ủy quyền. Tại đây, người dùng sẽ đăng nhập và chọn cung cấp quyền cho client.
3. **Bước 3: Authorization Server cấp mã Authorization Code**
   * Sau khi người dùng chấp thuận, Authorization Server sẽ gửi lại cho client một mã tạm thời gọi là **Authorization Code**.
4. **Bước 4: Client đổi Authorization Code lấy Access Token**
   * Client sẽ gửi **Authorization Code** này đến Authorization Server kèm theo một số thông tin xác thực (client ID và secret) để đổi lấy **Access Token**.
5. **Bước 5: Sử dụng Access Token để truy cập tài nguyên**
   * Sau khi có **Access Token**, client có thể sử dụng nó để truy cập tài nguyên trên Resource Server (ví dụ: truy cập thông tin người dùng qua API của Google).
6. **Bước 6: Làm mới Access Token (nếu cần)**
   * Nếu **Access Token** hết hạn, client có thể sử dụng một **Refresh Token** (nếu đã được cấp) để yêu cầu một **Access Token** mới mà không cần sự can thiệp từ người dùng.

### **Các loại Authorization Flows khác:**

Ngoài **Authorization Code Flow**, OAuth2 còn có các luồng khác phù hợp với từng loại ứng dụng khác nhau:

* **Implicit Flow**: Dùng cho ứng dụng phía client (Single Page Application), trong đó **Access Token** được cấp trực tiếp mà không qua bước trao đổi mã như Authorization Code Flow.
* **Resource Owner Password Credentials Flow**: Dùng trong trường hợp client hoàn toàn tin tưởng người dùng, như ứng dụng di động chính chủ.
* **Client Credentials Flow**: Dùng cho các dịch vụ server-to-server, không có sự can thiệp của người dùng.

### **Ưu điểm của OAuth2:**

* **Bảo mật cao**: Người dùng không cần chia sẻ mật khẩu trực tiếp với ứng dụng bên thứ ba.
* **Tính linh hoạt**: Có thể hỗ trợ nhiều loại ứng dụng khác nhau (web, mobile, server-side).
* **Quản lý quyền dễ dàng**: Người dùng có thể cấp hoặc thu hồi quyền truy cập của các ứng dụng một cách dễ dàng.