Implement Spring Security to manage the access to your application

Type: Projects

Skill: Spring Framework

Controller ImplementationSecurity ConfigurationSpring SecurityUser Model and RepositoryUserDetailsService Implementation

Medium

Develop a Spring Boot application that configures Spring Security to manage the access to your application.

**Tasks:**

**User Model and Repository:** Create a User entity with fields:

  1. **id** (type Long) : The unique id for the user.

  2. **username** (type String) : The username for the user.

  3. **password** (type String): The password for the user.

  4. **roles** (type String): The user's role.

The role type should be set as a string as user can have role "USER" or "ADMIN".

Implement a **UserRepository** interface that extends **JpaRepository** to manage the CRUD operations for the User entity.

**UserDetailsService Implementation:**

1. Implement a service class named **CustomUserDetailsService** that implements UserDetailsService interface.
2. Override the loadUserByUsername method to load a user by its username.

**Security Configuration:**

1. Implement a configuration class named **SecurityConfig** that extends **WebSecurityConfigurerAdapter.**
2. Override the configure(HttpSecurity http) method to define the security rules.
3. Configure HTTP Basic authentication.
4. Configure in-memory authentication with at least two users (one with USER role and another with ADMIN role).

**Controller Implementation:**

1. Implement a controller named HomeController using the **@RestController** annotation.

**Implement two endpoints:**

1. **GET /:** This endpoint should be accessible to all authenticated users and return a "Welcome" message.
2. **GET /admin:** This endpoint should be accessible only to users with the ADMIN role and return a "Welcome Admin" message.

**Test Cases:**

Your implementation will be evaluated against the following criteria:

1. **Basic Authentication:** The application should prompt for a username and password when trying to access any endpoint.
2. **User Access:** A user with the USER role should be able to access the / endpoint but not the /admin endpoint.
3. **Admin Access:** A user with the ADMIN role should be able to access both the / and /admin endpoints.

You are not required to implement a front-end for this application. It is sufficient to create the back-end API and test it using a tool like Postman.