**1. Cấu hình bảo mật**

import org.springframework.context.annotation.Bean;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter {

@Override

protected void configure(HttpSecurity http) throws Exception {

http

.csrf().disable()

.authorizeRequests()

.antMatchers("/api/login", "/api/register").permitAll()

.anyRequest().authenticated()

.and()

.addFilter(new JwtAuthenticationFilter(authenticationManager()))

.addFilter(new JwtAuthorizationFilter(authenticationManager()));

}

}

**2. Tạo API Đăng nhập**

@RestController

@RequestMapping("/api")

public class AuthController {

@PostMapping("/login")

public ResponseEntity<?> login(@RequestBody LoginRequest loginRequest) {

// Xác thực người dùng

// Tạo token JWT

String token = jwtTokenProvider.generateToken(loginRequest.getUsername());

return ResponseEntity.ok(new JwtResponse(token));

}

}

**3. Tạo API Đăng ký**

@PostMapping("/register")

public ResponseEntity<?> register(@RequestBody RegisterRequest registerRequest) {

// Kiểm tra và tạo tài khoản người dùng

// Trả về thông báo thành công

return ResponseEntity.ok("Đăng ký thành công");

}

**4. API Đăng xuất**

Với JWT, bạn không cần thực hiện gì ở phía server. Để đăng xuất, bạn chỉ cần xóa token ở phía client.

**5. API kiểm tra token**

@GetMapping("/test-token")

public ResponseEntity<?> testToken(HttpServletRequest request) {

String token = request.getHeader("Authorization");

if (token == null || !jwtTokenProvider.validateToken(token)) {

return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Lỗi: Không có token hoặc token không hợp lệ");

}

return ResponseEntity.ok("Thành công");

}

**6. Tạo JwtTokenProvider để tạo và xác thực token JWT**

@Component

public class JwtTokenProvider {

// Secret key và các phương thức để tạo và xác thực JWT

public String generateToken(String username) {

// Tạo token

}

public boolean validateToken(String token) {

// Kiểm tra tính hợp lệ của token

}

}