//AuthenticationController.java

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.util.JwtUtil;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import javax.servlet.http.HttpServletRequest;

import java.util.Base64;

@RestController

public class AuthenticationController {

@Autowired

private JwtUtil jwtUtil;

@RequestMapping(value = "/authenticate", method = RequestMethod.GET)

public ResponseEntity<?> authenticate(HttpServletRequest request) {

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

if (authHeader != null && authHeader.startsWith("Basic ")) {

String[] credentials = extractCredentials(authHeader);

String username = credentials[0];

String password = credentials[1];

// For demo: simple hardcoded check

if ("user".equals(username) && "pwd".equals(password)) {

String token = jwtUtil.generateToken(username);

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

}

}

return ResponseEntity.status(401).build(); // Unauthorized

}

private String[] extractCredentials(String authHeader) {

String base64Credentials = authHeader.substring("Basic ".length());

String decoded = new String(Base64.getDecoder().decode(base64Credentials));

return decoded.split(":", 2);

}

static class JwtResponse {

private String token;

public JwtResponse(String token) { this.token = token; }

public String getToken() { return token; }

public void setToken(String token) { this.token = token; }

}

}

//SecurityConfig.java

package com.cognizant.springlearn.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

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

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http.csrf().disable()

.authorizeHttpRequests()

.requestMatchers("/authenticate").permitAll()

.anyRequest().authenticated()

.and().httpBasic(); // Enable HTTP Basic Auth

return http.build();

}

}

//JwtUtil.java

package com.cognizant.springlearn.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.springframework.stereotype.Component;

import java.util.Date;

@Component

public class JwtUtil {

private final String SECRET\_KEY = "mySecretKey"; // Keep it safe!

public String generateToken(String username) {

long currentTimeMillis = System.currentTimeMillis();

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date(currentTimeMillis))

.setExpiration(new Date(currentTimeMillis + 1000 \* 60 \* 10)) // 10 min

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

}

}



