**Create authentication service that returns JWT**

**SecurityConfig.java:**

package 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 securityFilterChain(HttpSecurity http) throws Exception {  
 http  
 .csrf(csrf -> csrf.disable())  
 .authorizeHttpRequests(auth -> auth  
 .requestMatchers("/authenticate").permitAll()  
 .anyRequest().authenticated()  
 )  
 .httpBasic(customizer -> {}); *// Updated to fix deprecation* return http.build();  
 }  
}

**AuthController.java:**

package controller;  
import jakarta.servlet.http.HttpServletRequest;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
import util.JwtUtil;  
  
import java.util.Base64;  
import java.util.HashMap;  
import java.util.Map;  
  
@RestController  
public class AuthController {  
  
 @Autowired  
 private JwtUtil jwtUtil;  
  
 @GetMapping("/authenticate")  
 public ResponseEntity<?> authenticate(HttpServletRequest request) {  
 String header = request.getHeader("Authorization");  
  
 if (header == null || !header.startsWith("Basic ")) {  
 return ResponseEntity.*status*(401).body("Missing or invalid Authorization header");  
 }  
  
 try {  
 String base64Credentials = header.substring("Basic ".length());  
 String decoded = new String(Base64.*getDecoder*().decode(base64Credentials));  
 String[] credentials = decoded.split(":", 2);  
  
 if (credentials.length != 2) {  
 return ResponseEntity.*status*(400).body("Invalid credentials format");  
 }  
  
 String username = credentials[0];  
 String password = credentials[1];  
  
 if ("user".equals(username) && "pwd".equals(password)) {  
 String token = jwtUtil.generateToken(username);  
  
 Map<String, String> response = new HashMap<>();  
 response.put("token", token);  
  
 return ResponseEntity.*ok*(response);  
 } else {  
 return ResponseEntity.*status*(401).body("Invalid username or password");  
 }  
  
 } catch (IllegalArgumentException e) {  
 return ResponseEntity.*status*(400).body("Malformed Base64 encoding");  
 }  
 }  
}

**AuthserviceApplication.java:**

package com.example.authservice;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class AuthserviceApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(AuthserviceApplication.class, args);  
 }  
  
}

**JwtUtil.java:**

package 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 = "secretKey123456"; *// Use a secure key in production* public String generateToken(String username) {  
 return Jwts.*builder*()  
 .setSubject(username)  
 .setIssuedAt(new Date(System.*currentTimeMillis*()))  
 .setExpiration(new Date(System.*currentTimeMillis*() + 1000 \* 60 \* 10)) *// 10 minutes* .signWith(SignatureAlgorithm.*HS256*, secret.getBytes())  
 .compact();  
 }  
}

