**Create authentication service that returns JWT**

AuthenticationController.java

**package** com.cognizant.spring\_learn.controller;

**import** jakarta.servlet.http.HttpServletRequest;

**import** org.springframework.http.ResponseEntity;

**import** java.util.Base64;

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

**import** com.cognizant.spring\_learn.service.AuthenticationService;

**import** java.nio.charset.StandardCharsets;

@RestController

**public** **class** AuthenticationController {

**private** **final** AuthenticationService service;

**public** AuthenticationController(AuthenticationService service) {

**this**.service = service;

}

@GetMapping("/authenticate")

**public** ResponseEntity<?> authenticate(HttpServletRequest request) {

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

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

**return** ResponseEntity.*status*(401).body("Missing or invalid Authorization header");

}

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

**byte**[] credDecoded = Base64.*getDecoder*().decode(base64Credentials);

String credentials = **new** String(credDecoded, StandardCharsets.***UTF\_8***);

String[] values = credentials.split(":", 2);

String username = values[0];

String password = values[1];

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

String token = service.generateToken(username);

**return** ResponseEntity.*ok*().body("{\"token\":\"" + token + "\"}");

} **else** {

**return** ResponseEntity.*status*(401).body("Invalid credentials");

}

}

}

SecurityConfig.java

**package** com.cognizant.spring\_learn.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.core.userdetails.User;

**import** org.springframework.security.core.userdetails.UserDetailsService;

**import** org.springframework.security.provisioning.InMemoryUserDetailsManager;

**import** org.springframework.security.web.SecurityFilterChain;

**import** org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

**import** org.springframework.security.crypto.password.PasswordEncoder;

@Configuration

**public** **class** SecurityConfig {

@Bean

**public** SecurityFilterChain filterChain(HttpSecurity http) **throws** Exception {

http

.csrf(csrf -> csrf.disable())

.authorizeHttpRequests(auth -> auth

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

.anyRequest().authenticated()

)

.~~httpBasic~~();

**return** http.build();

}

@Bean

**public** PasswordEncoder passwordEncoder() {

**return** **new** BCryptPasswordEncoder();

}

@Bean

**public** UserDetailsService users(PasswordEncoder encoder) {

**return** **new** InMemoryUserDetailsManager(

User.*withUsername*("user")

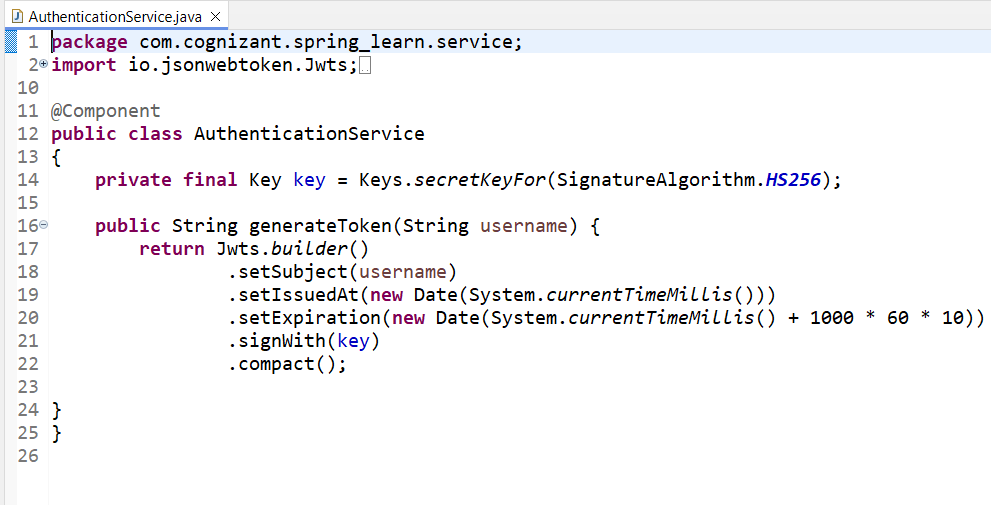
.password(encoder.encode("pwd"))

.roles("USER")

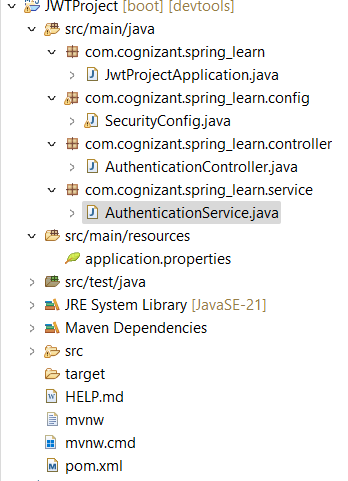
.build()

);

}

}

Project Structure:



"token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzUyNDA5NTU1LCJleHAiOjE3NTI0MTAxNTV9.Dxmf6WDX4cOrv2\_LMJgHFqwh6twVhBL3JmLJe-KLats"A screenshot of a computer

Description automatically generated