**JWT**

**Exercise 1: Create authentication service that returns JWT**

**Code:**

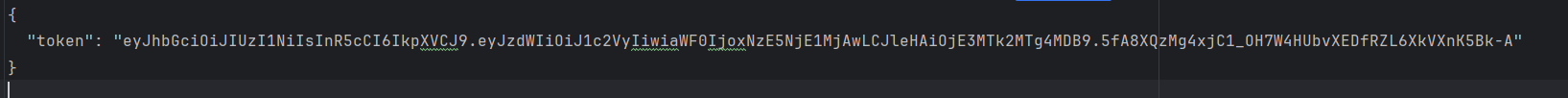
@Component  
public class JwtUtil {  
 private final Key key = Keys.*secretKeyFor*(SignatureAlgorithm.*HS256*);  
  
 public String generateToken(String username){  
 return Jwts.*builder*()  
 .setSubject(username)  
 .setIssuedAt(new Date(System.*currentTimeMillis*()))  
 .setExpiration(new Date(System.*currentTimeMillis*() + 1000 \* 60 \* 60)) // 1 hour  
 .signWith(key)  
 .compact();  
 }  
}

@RestController  
public class AuthenticationController {  
 private final JwtUtil jwtUtil;  
  
 public AuthenticationController(JwtUtil jwtUtil) {  
 this.jwtUtil = jwtUtil;  
 }  
  
 @GetMapping("/authenticate")  
 public ResponseEntity<?> authenticate(@RequestHeader("Authorization") String authHeader) {  
 if (authHeader != null && authHeader.startsWith("Basic ")) {  
 String base64Credentials = authHeader.substring("Basic ".length());  
 byte[] credDecoded = Base64.*getDecoder*().decode(base64Credentials);  
 String credentials = new String(credDecoded);  
 String[] values = credentials.split(":", 2);  
  
 String username = values[0];  
 String password = values[1];  
  
 if (username.equals("user") && password.equals("pwd")) {  
 String token = jwtUtil.generateToken(username);  
 return ResponseEntity.*ok*().body("{\"token\":\"" + token + "\"}");  
 } else {  
 return ResponseEntity.*status*(401).body("Invalid credentials");  
 }  
 }  
 return ResponseEntity.*badRequest*().body("Missing or invalid Authorization header");  
 }  
}

@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()  
 );  
 return http.build();  
 }  
}

<dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-api</artifactId>  
 <version>0.11.5</version>  
 </dependency>  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-impl</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-jackson</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>  
  
</dependencies>

**Output:**

****