**Create authentication service that returns JWT**   
  
**CODE**:

package com.cognizant.springlearn.controller;  
  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
import io.jsonwebtoken.security.Keys;  
import jakarta.servlet.http.HttpServletRequest;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
  
import java.security.Key;  
import java.util.Base64;  
import java.util.Collections;  
import java.util.Date;  
  
  
@RestController  
public class AuthenticationController {  
  
 private static final Key *key* = Keys.*secretKeyFor*(SignatureAlgorithm.*HS256*); // ✅ SAFE KEY  
  
 @GetMapping("/authenticate")  
 public ResponseEntity<?> authenticate(@RequestHeader("Authorization") String authHeader) {  
 // Basic Auth decoding  
 String base64Credentials = authHeader.substring("Basic ".length());  
 byte[] decoded = Base64.*getDecoder*().decode(base64Credentials);  
 String[] credentials = new String(decoded).split(":");  
 String username = credentials[0];  
 String password = credentials[1];  
  
 if ("user".equals(username) && "pwd".equals(password)) {  
 String jwt = Jwts.*builder*()  
 .setSubject(username)  
 .signWith(*key*)  
 .compact();  
 return ResponseEntity.*ok*(Collections.*singletonMap*("token", jwt));  
 } else {  
 return ResponseEntity.*status*(HttpStatus.*UNAUTHORIZED*).build();  
 }  
 }  
}

package com.cognizant.springlearn.config;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.Customizer;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
import org.springframework.security.config.annotation.web.configurers.AbstractHttpConfigurer;  
  
@Configuration  
public class SecurityConfig {  
  
 @Bean  
 public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {  
 http  
 .csrf(AbstractHttpConfigurer::disable)  
 .authorizeHttpRequests(auth -> auth  
 .requestMatchers("/authenticate").permitAll() // ✅ Open token endpoint  
 .anyRequest().authenticated() // ✅ Protect everything else  
 )  
 .httpBasic(Customizer.*withDefaults*()); // ✅ Replaces deprecated .httpBasic()  
  
 return http.build();  
 }  
}

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.5.3</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>spring-learn</name>  
 <description>Example project for Spring Boot REST</description>  
 <url/>  
 <licenses>  
 <license/>  
 </licenses>  
 <developers>  
 <developer/>  
 </developers>  
 <scm>  
 <connection/>  
 <developerConnection/>  
 <tag/>  
 <url/>  
 </scm>  
 <properties>  
 <java.version>21</java.version>  
 </properties>  
 <dependencies>  
  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-api</artifactId>  
 <version>0.11.5</version>  
 </dependency>  
  
 <!-- JWT Implementation -->  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-impl</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>  
  
 <!-- JWT JSON Serializer (Jackson) -->  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt-jackson</artifactId>  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>io.jsonwebtoken</groupId>  
 <artifactId>jjwt</artifactId>  
 <version>0.9.1</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

application.properties:

spring.application.name=spring-learn  
  
  
server.port=8090  
logging.level.com.cognizant.springlearn=DEBUG  
  
spring.security.user.name=user  
spring.security.user.password=pwd

**OUTPUT:**

