**Create authentication service that returns JWT**

**SecurityConfig.java**

package com.affu.jwt.authservice.config;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
  
@Configuration  
@EnableWebSecurity  
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(); // Basic authentication  
 return http.build();  
 }  
}

**AuthController.java**

package com.affu.jwt.authservice.controller;  
  
import com.affu.jwt.authservice.util.JwtUtil;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import javax.servlet.http.HttpServletRequest;  
import java.util.Base64;  
  
@RestController  
public class AuthController {  
  
 @GetMapping("/authenticate")  
 public ResponseEntity<?> authenticate(HttpServletRequest request) {  
 String authHeader = request.getHeader("Authorization");  
  
 if (authHeader != null && authHeader.startsWith("Basic ")) {  
 String base64Credentials = authHeader.substring("Basic ".length());  
 byte[] decodedBytes = Base64.*getDecoder*().decode(base64Credentials);  
 String credentials = new String(decodedBytes);  
 String[] parts = credentials.split(":", 2);  
  
 if (parts.length == 2) {  
 String username = parts[0];  
 String password = parts[1];  
  
 if ("user".equals(username) && "pwd".equals(password)) {  
 String token = JwtUtil.*generateToken*(username);  
 return ResponseEntity.ok().body("{\"token\":\"" + token + "\"}");  
 }  
 }  
 }  
  
 return ResponseEntity.status(401).body("Invalid credentials");  
 }  
}

**JwtUtil.java**

package com.affu.jwt.authservice.util;  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
  
import java.util.Date;  
public class JwtUtil {  
  
 private static final String *SECRET\_KEY* = "secretkey";  
 private static final long *EXPIRATION\_TIME* = 60 \* 60 \* 1000; // 1 hour  
  
 public static String generateToken(String username) {  
 return Jwts.builder()  
 .setSubject(username)  
 .setIssuedAt(new Date())  
 .setExpiration(new Date(System.*currentTimeMillis*() + *EXPIRATION\_TIME*))  
 .signWith(SignatureAlgorithm.HS256, *SECRET\_KEY*.getBytes())  
 .compact();  
 }  
}

**AuthserviceApplication.java**

package com.affu.jwt.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);  
 }  
}

**Applications properties**

spring.application.name=authservice  
server.port=8090

**Pom.xml**

<?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>  
  
 <groupId>com.affu</groupId>  
 <artifactId>authservice</artifactId>  
 <version>1.0.0</version>  
 <name>authservice</name>  
 <description>JWT Authentication Service</description>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.15</version>  
 <relativePath/>  
 </parent>  
  
 <properties>  
 <java.version>11</java.version>   
 </properties>  
  
 <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>io.jsonwebtoken</groupId>  
 <artifactId>jjwt</artifactId>  
 <version>0.9.1</version>  
 </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>  
 <!-- Spring Boot Maven Plugin -->  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <configuration>  
 <excludes>  
 <exclude>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 </exclude>  
 </excludes>  
</configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**OUTPUT**

