**Create authentication service that returns JWT**

**AuthenticationController.java**

**Path:** src/main/java/com/cognizant/jwt/AuthenticationController.java

**CODE :-**

**package** com.cognizant.jwt;

**import** io.jsonwebtoken.Jwts;

**import** io.jsonwebtoken.security.Keys;

**import** io.jsonwebtoken.SignatureAlgorithm;

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

**import** org.springframework.security.core.Authentication;

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

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

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

**import** java.security.Key;

**import** java.util.Date;

@RestController

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

**private** **static** **final** String ***SECRET\_KEY*** = "secret-key-which-is-at-least-32-characters!";

@GetMapping("/authenticate")

**public** ResponseEntity<?> authenticate(Authentication authentication) {

String username = authentication.getName();

Key key = Keys.*hmacShaKeyFor*(***SECRET\_KEY***.getBytes(StandardCharsets.***UTF\_8***));

String token = Jwts.*builder*()

.subject(username)

.issuedAt(**new** Date())

.expiration(**new** Date(System.*currentTimeMillis*() + 10 \* 60 \* 1000)) // 10 mins

.signWith(key, SignatureAlgorithm.HS256)

.compact();

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

}

}

**SecurityConfig.java**

**Path:** src/main/java/com/cognizant/jwt/SecurityConfig.java

**CODE :-**

**package** com.cognizant.jwt;

**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;

@Configuration

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

@Bean

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

http

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

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").authenticated()

.anyRequest().permitAll()

)

.httpBasic();

**return** http.build();

}

@Bean

**public** UserDetailsService userDetailsService() {

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

User.*withUsername*("user")

.password("{noop}pwd")

.roles("USER")

.build()

);

}

}

**JwtAuthenticationApplication.java**

**Path:** src/main/java/com/cognizant/jwt/JwtAuthenticationApplication.java

**CODE :-**

**package com.cognizant.jwt;**

**import org.springframework.boot.SpringApplication;**

**import org.springframework.boot.autoconfigure.SpringBootApplication;**

**@SpringBootApplication**

**public class JwtAuthenticationApplication {**

**public static void main(String[] args) {**

**SpringApplication.*run*(JwtAuthenticationApplication.class, args);**

**}**

**}**

**application.properties**

**Path:** src/main/resources/application.properties

**CODE :-**

server.port=8090

spring.security.user.name=user

spring.security.user.password=pwd

spring.application.name=jwt-authentication

**pom.xml**

**CODE :-**

<?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>3.4.7</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.cognizant</groupId>

<artifactId>jwt-authentication</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>jwt-authentication</name>

<description>Spring Boot JWT Authentication Service</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>21</java.version>

</properties>

<dependencies>

<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-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-api</artifactId>

<version>0.12.5</version>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-impl</artifactId>

<version>0.12.5</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-jackson</artifactId>

<version>0.12.5</version>

<scope>runtime</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**OUTPUT :-**





