**Create authentication service that returns JWT:**

**//JwtApplication.java**

package com.cognizant.jwt;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class JwtApplication {

public static void main(String[] args) {

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

}

}

**//AuthenticationController.java**

package com.cognizant.jwt.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import java.util.Base64;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import io.jsonwebtoken.security.Keys;

import java.security.Key;

import java.nio.charset.StandardCharsets;

*@RestController*

public class AuthenticationController {

private static final Logger *LOGGER* = LoggerFactory.*getLogger*(AuthenticationController.class);

private static final String *SECRET* = "my-secret-key-that-is-at-least-32-characters-long";

*@GetMapping*("/authenticate")

public Map<String, String> authenticate(*@RequestHeader*("Authorization") String authHeader) {

*LOGGER*.info("Start authenticate");

*LOGGER*.debug("Authorization Header: {}", authHeader);

Map<String, String> map = new HashMap<>();

String user = getUser(authHeader);

*LOGGER*.debug("Username: {}", user);

String token = generateJwt(user);

map.put("token", token);

*LOGGER*.info("End authenticate");

return map;

}

private String getUser(String authHeader) {

*LOGGER*.info("Start getUser");

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

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

String credentials = new String(credDecoded);

*LOGGER*.debug("Decoded credentials: {}", credentials);

String username = credentials.split(":", 2)[0];

*LOGGER*.info("End getUser");

return username;

}

private String generateJwt(String user) {

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

return Jwts.*builder*()

.setSubject(user)

.setIssuedAt(new Date())

.setExpiration(new Date(System.*currentTimeMillis*() + 20 \* 60 \* 1000))

.signWith(key, *SignatureAlgorithm*.*HS256*)

.compact();

}

}

**//SecurityConfig.java**

package com.cognizant.jwt.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.configuration.EnableWebSecurity;

import org.springframework.security.web.SecurityFilterChain;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.core.userdetails.User;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

*@Configuration*

*@EnableWebSecurity*

public class SecurityConfig {

*@Bean*

public InMemoryUserDetailsManager userDetailsService() {

UserDetails user = User

.*withUsername*("user")

.password("{noop}pwd")

.roles("USER")

.build();

return new InMemoryUserDetailsManager(user);

}

*@Bean*

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").hasAnyRole("USER", "ADMIN")

.anyRequest().authenticated()

)

.httpBasic(Customizer.*withDefaults*())

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

return http.build();

}

}

**//application.properties**

spring.application.name=jwt

server.port=8090

logging.level.com.cognizant.jwt=DEBUG

logging.level.com.cognizant.jwt=TRACE

**//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>

<parent>

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

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.5.3</version>

<relativePath/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>jwt</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>jwt</name>

<description>Demo project for Spring Boot</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-web</artifactId>

</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>

<dependency>

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

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<dependency>

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

<artifactId>spring-boot-starter</artifactId>

</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>

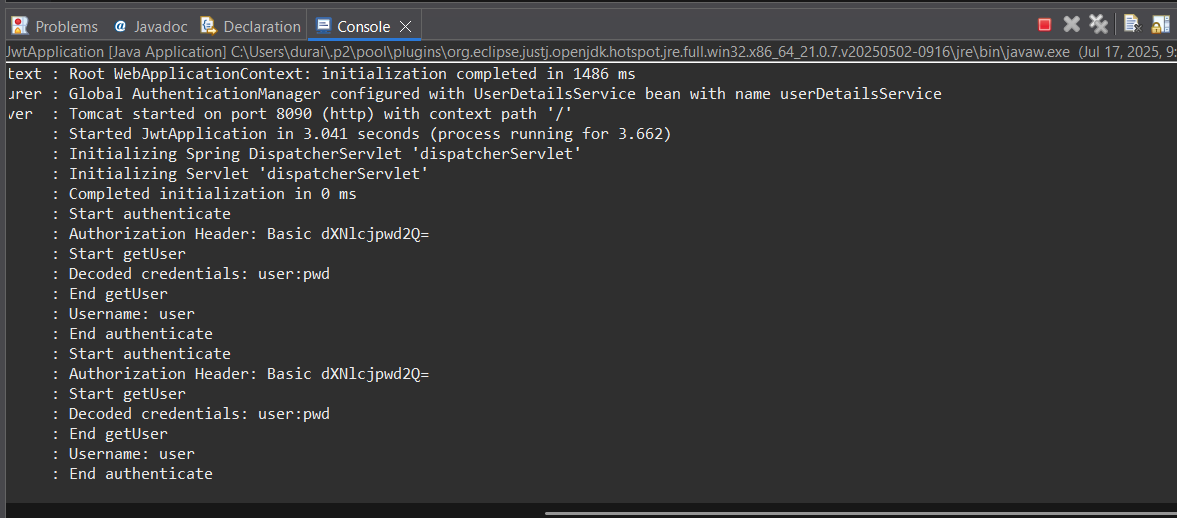
</plugin>

</plugins>

</build>

</project>

**Output:**

****

****