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

**SecurityConfig.java:**

**package** com.cognizant.spring\_learn.config;

**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.web.SecurityFilterChain;

**import** org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

@Configuration

@EnableWebSecurity

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

@Bean

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

http.csrf().disable()

.authorizeHttpRequests()

.anyRequest().permitAll();

**return** http.build();

}

}

**AuthenticationController.java:**

**package** com.cognizant.spring\_learn.controller;

**import** com.cognizant.spring\_learn.util.JwtUtil;

**import** org.springframework.http.HttpStatus;

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

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

**import** java.util.Base64;

**import** java.util.Collection;

**import** java.util.Collections;

@RestController

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

@GetMapping("/authenticate")

**public** ResponseEntity<?> authenticate(@RequestHeader("Authorization")String authHeader){

**if**(authHeader==**null**||!authHeader.startsWith("Basic ")) {

**return** ResponseEntity.*status*(HttpStatus.***UNAUTHORIZED***).body("Missing or invalid Authorization header");

}

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

String credentials=**new** String(Base64.*getDecoder*().decode(base64Credentials));

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*(Collections.*singletonMap*("token",token));

}

**else** {

**return** ResponseEntity.*status*(HttpStatus.***UNAUTHORIZED***).body("Invalid Credentials");

}

}

}

**JwtUtil.java:**

**package** com.cognizant.spring\_learn.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***=1000\*60\*10;

**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***)

.compact();

}

}

**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>2.7.18</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>Demo project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

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

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

**Output:**



