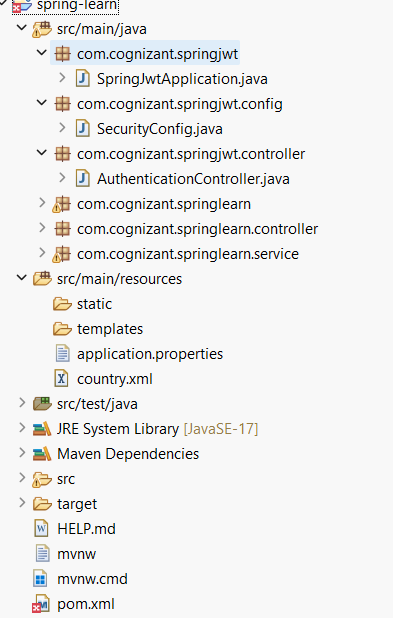
**1. Create authentication service that returns JWT**

**Project Name: Spring-learn**

**Project Structure:**



**Code:**

AuthenticationController.java

package com.cognizant.springjwt.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.springframework.http.HttpStatus;

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

import org.springframework.web.server.ResponseStatusException;

import javax.crypto.SecretKey;

import java.util.Date;

@RestController

@RequestMapping("/api")

public class AuthenticationController {

private static final SecretKey SECRET\_KEY =

Keys.hmacShaKeyFor("MySecretKey1234567890MySecretKey1234567890".getBytes());

@PostMapping("/authenticate")

public TokenResponse authenticate(@RequestBody AuthRequest authRequest) {

if (!"user".equals(authRequest.getUsername()) || !"pass".equals(authRequest.getPassword())) {

throw new ResponseStatusException(HttpStatus.UNAUTHORIZED, "Invalid credentials");

}

String token = Jwts.builder()

.setSubject(authRequest.getUsername())

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 3600000)) // 1 hour

.signWith(SECRET\_KEY, SignatureAlgorithm.HS256)

.compact();

return new TokenResponse(token);

}

static class AuthRequest {

private String username;

private String password;

public String getUsername() { return username; }

public void setUsername(String username) { this.username = username; }

public String getPassword() { return password; }

public void setPassword(String password) { this.password = password; }

}

static class TokenResponse {

private String token;

public TokenResponse(String token) { this.token = token; }

public String getToken() { return token; }

public void setToken(String token) { this.token = token; }

}

}

SecurityConfig.java

package com.cognizant.springjwt.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;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

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

.authorizeHttpRequests(auth -> auth

.requestMatchers("/api/authenticate").permitAll()

.anyRequest().authenticated()

)

.httpBasic(httpBasic -> {}); // optional

return http.build();

}

}

application.properties

server.port=8090

spring.application.name=spring-learn

logging.level.com.cognizant.springjwt=DEBUG

**Output:**



