**Create authentication service that returns JWT**

**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.Customizer;

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 UserDetailsService users() {

        return new InMemoryUserDetailsManager(

                User.withUsername("user")

                    .password("{noop}pwd")      // {noop} = no encoder

                    .roles("USER")

                    .build());

    }

    @Bean

    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

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

            .authorizeHttpRequests(auth -> auth

                    .requestMatchers("/authenticate", "/authenticate/\*\*").permitAll()

                    .anyRequest().authenticated())

            .httpBasic(Customizer.withDefaults());   // enable Basic auth

        return http.build();

    }

}

**AuthenticationController.java:**

package com.cognizant.spring\_learn.controller;

import java.util.Base64;

import java.util.Collections;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

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

import org.springframework.web.bind.annotation.RequestHeader;

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

import org.springframework.web.server.ResponseStatusException;

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

@RestController

public class AuthenticationController {

    @Autowired

    private JwtUtil jwtUtil;

    @GetMapping("/authenticate")

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

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

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

        String[] credentials = new String(decodedBytes).split(":", 2);

        String username = credentials[0];

        String password = credentials[1];

        if (!username.equals("user") || !password.equals("pwd")) {

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

        }

        String token = jwtUtil.generateToken(username);

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

    }

}

**JwtUtil.java:**

package com.cognizant.spring\_learn.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import java.security.Key;

import java.util.Date;

import org.springframework.stereotype.Component;

@Component

public class JwtUtil {

    private final Key key = Keys.secretKeyFor(SignatureAlgorithm.HS256);

    public String generateToken(String username) {

        return Jwts.builder()

                .setSubject(username)

                .setIssuedAt(new Date())

                .setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 60 \* 10)) // 10 hours

                .signWith(key)

                .compact();

    }

}

**Output:**

