**Create authentication service that returns JWT**

AuthenticationController Code:

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

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

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

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

**import** java.util.Base64;

@RestController

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

**private** **final** JwtUtil jwtUtil;

**public** AuthenticationController(JwtUtil jwtUtil) {

**this**.jwtUtil = jwtUtil;

}

@GetMapping("/ping")

**public** String ping() {

System.***out***.println("Ping endpoint hit");

**return** "pong";

}

@RequestMapping(value = "/authenticate", method = {RequestMethod.***GET***, RequestMethod.***POST***})

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

System.***out***.println("🔒 /authenticate endpoint called");

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

System.***out***.println("❌ Missing/Invalid Authorization header");

**return** ResponseEntity.*status*(401).body("Missing or invalid Authorization header");

}

// Decode Base64 credentials

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

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

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

String username = credentials[0];

String password = credentials[1];

System.***out***.println("Username: " + username);

System.***out***.println("Password: " + password);

**if** ("user".equals(username) && "pwd".equals(password)) {

String token = jwtUtil.generateToken(username);

System.***out***.println("Token generated");

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

} **else** {

System.***out***.println("Invalid credentials");

**return** ResponseEntity.*status*(401).body("Invalid credentials");

}

}

}

JwtUtil Code:

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

**import** org.springframework.stereotype.Component;

**import** io.jsonwebtoken.Jwts;

**import** io.jsonwebtoken.SignatureAlgorithm;

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

**import** java.util.Date;

**import** java.security.Key;

@Component

**public** **class** JwtUtil {

**private** String secret = "your\_256\_bit\_secret\_string\_which\_should\_be\_very\_long";

**private** Key key = Keys.*hmacShaKeyFor*(secret.getBytes());

**public** String generateToken(String username) {

**return** Jwts.*builder*()

.setSubject(username)

.setIssuedAt(**new** Date())

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

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

.compact();

}

}

Output:

