**JWT-handson**

**Create authentication service that returns JWT**

**pom.xml**

<dependencies>

<!-- Spring Web -->

<dependency>

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

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

</dependency>

<!-- Spring Security -->

<dependency>

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

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

</dependency>

<!-- JJWT: JWT library -->

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

</dependencies>

**Authentication Controller**

package com.cognizant.springlearn.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.springframework.http.ResponseEntity;

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

import javax.servlet.http.HttpServletRequest;

import java.util.Base64;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

@RestController

public class AuthenticationController {

private static final String SECRET\_KEY = "your\_secret\_key"; // keep it safe and private

private static final long EXPIRATION\_TIME = 10 \* 60 \* 1000; // 10 minutes

@RequestMapping("/authenticate")

public ResponseEntity<?> authenticate(HttpServletRequest request) {

String authHeader = request.getHeader("Authorization");

// Check if header is missing or not in Basic auth format

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

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

}

// Decode the header to get username:password

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

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

String credentials = new String(credDecoded);

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

if (parts.length != 2) {

return ResponseEntity.status(400).body("Invalid Basic Auth format");

}

String username = parts[0];

String password = parts[1];

// For demo purposes: hardcoded user validation

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

return ResponseEntity.status(403).body("Invalid Credentials");

}

// Generate JWT

String token = Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date(System.currentTimeMillis()))

.setExpiration(new Date(System.currentTimeMillis() + EXPIRATION\_TIME))

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

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

response.put("token", token);

return ResponseEntity.ok(response);

}

}

**SecurityConfig.java**

package com.cognizant.springlearn.config;

import org.springframework.context.annotation.Configuration;

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

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

@Configuration

@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter {

@Override

protected void configure(HttpSecurity http) throws Exception {

http

.csrf().disable() // disable CSRF for simplicity

.authorizeRequests()

.antMatchers("/authenticate").permitAll() // allow public access to /authenticate

.anyRequest().authenticated()

.and()

.httpBasic(); // enable Basic auth

}

}

**Configure Port and Logging**

server.port=8090

logging.level.org.springframework=info

logging.level.com.cognizant=debug

**Output:**

{

"token": "eyJhbGciOiJIUzI1NiJ9..."

}