Harshitha M

Superset id : 6390499

WEEK 4

Spring Boot JWT Authentication - Hands-On

# Create Authentication Service that Returns JWT

## Steps:

* 1. Create a new controller named AuthenticationController in package com.cognizant.spring-learn.controller.
* 2. Add endpoint '/authenticate' with method GET and annotate with @GetMapping.
* 3. Accept user credentials using HTTP Basic Authentication.
* 4. In the method, decode the Authorization header using Base64 decoding to extract username and password.
* 5. Generate a JWT using io.jsonwebtoken.Jwts and return the token in response.
* 6. Configure Spring Security to allow '/authenticate' as a permitted path in SecurityConfig.
* 7. Test the endpoint using curl:  
   curl -s -u user:pwd http://localhost:8090/authenticate
* 8. Validate that the response includes a valid JWT token.

## AuthenticationController.java:

@RestController  
public class AuthenticationController {  
  
 @GetMapping("/authenticate")  
 public Map<String, String> authenticate(@RequestHeader("Authorization") String authHeader) {  
 System.out.println("START - authenticate()");  
  
 String base64Credentials = authHeader.substring("Basic ".length());  
 byte[] credDecoded = Base64.getDecoder().decode(base64Credentials);  
 String credentials = new String(credDecoded, StandardCharsets.UTF\_8);  
 String[] values = credentials.split(":", 2);  
 String username = values[0];  
 String password = values[1];  
  
 System.out.println("Credentials received - Username: " + username);  
  
 // Simulate validation (normally check from DB or service)  
 if ("user".equals(username) && "pwd".equals(password)) {  
 String token = Jwts.builder()  
 .setSubject(username)  
 .setIssuedAt(new Date())  
 .setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 10))  
 .signWith(SignatureAlgorithm.HS256, "secretkey")  
 .compact();  
  
 Map<String, String> response = new HashMap<>();  
 response.put("token", token);  
  
 System.out.println("END - authenticate() : Token generated");  
 return response;  
 } else {  
 throw new ResponseStatusException(HttpStatus.UNAUTHORIZED, "Invalid credentials");  
 }  
 }  
}

## SecurityConfig.java:

@Configuration  
@EnableWebSecurity  
public class SecurityConfig extends WebSecurityConfigurerAdapter {  
  
 @Override  
 protected void configure(HttpSecurity http) throws Exception {  
 http.csrf().disable()  
 .authorizeRequests()  
 .antMatchers("/authenticate").permitAll()  
 .anyRequest().authenticated()  
 .and()  
 .httpBasic();  
 }  
}

## Test with curl:

curl -s -u user:pwd http://localhost:8090/authenticate

OUTPUT:

A screen shot of a computer

AI-generated content may be incorrect.