**3. Create Authentication Service That Returns JWT**

**a) Add Dependencies to pom.xml**

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

<dependency>

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

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

</dependency>

**b) JWT Utility**

package com.example.rest.security;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import java.util.Date;

public class JwtUtil {

private static final String SECRET\_KEY = "mySecretKey";

private static final long EXPIRATION\_TIME = 86400000; // 1 day

public static String generateToken(String username) {

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

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

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

}

}

**c) Authentication Controller**

package com.example.rest.controller;

import com.example.rest.security.JwtUtil;

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

import java.util.HashMap;

import java.util.Map;

@RestController

@RequestMapping("/auth")

public class AuthController {

@PostMapping("/login")

public Map<String, String> login(@RequestParam String username, @RequestParam String password) {

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

if ("admin".equals(username) && "password".equals(password)) {

String token = JwtUtil.generateToken(username);

response.put("token", token);

} else {

response.put("error", "Invalid credentials");

}

return response;

}

}

**d) Spring Security Configuration (Optional for Token Protection)**

package com.example.rest.config;

import org.springframework.context.annotation.\*;

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().disable()

.authorizeRequests().anyRequest().permitAll();

return http.build();

}

}

**e) Test Login Endpoint**

Request:

POST /auth/login

Body: username=admin&password=password

Response:

{

"token": "eyJhbGciOiJIUzI1NiJ9..."

}