# SKILL - Spring REST using Spring Boot 3

# Create authentication service that returns JWT

# 1. Purpose

To create a REST-based authentication service in Spring Boot that authenticates user credentials and returns a JWT token. The credentials are passed using HTTP Basic Auth.

# 2. JwtUtil.java

package com.cognizant.auth.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.springframework.stereotype.Component;

import java.util.Date;

@Component

public class JwtUtil {

private String secret = "mySecretKey";

public String generateToken(String username) {

long currentTimeMillis = System.currentTimeMillis();

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date(currentTimeMillis))

.setExpiration(new Date(currentTimeMillis + 1000 \* 60 \* 10)) // Token valid for 10 minutes

.signWith(SignatureAlgorithm.HS256, secret)

.compact();

}

}

# 3. AuthController.java

package com.cognizant.auth.controller;

import com.cognizant.auth.util.JwtUtil;

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

import org.springframework.http.ResponseEntity;

import org.springframework.util.Base64Utils;

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

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

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

@RestController

public class AuthController {

@Autowired

private JwtUtil jwtUtil;

@GetMapping("/authenticate")

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

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

String credentials = new String(Base64Utils.decodeFromString(base64Credentials));

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

String username = values[0];

String password = values[1];

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

String token = jwtUtil.generateToken(username);

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

} else {

return ResponseEntity.status(401).body("Invalid Credentials");

}

}

}

# 4. SecurityConfig.java

package com.cognizant.auth.config;

import org.springframework.context.annotation.Configuration;

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

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

@Configuration

public class SecurityConfig extends WebSecurityConfigurerAdapter {

@Override

protected void configure(HttpSecurity http) throws Exception {

http

.csrf().disable()

.authorizeRequests()

.antMatchers("/authenticate").permitAll()

.anyRequest().authenticated()

.and()

.httpBasic();

}

}

# 5.Main:

@SpringBootApplication

public class SpringBootJwtApplication {

public static void main(String[] args) {

SpringApplication.run(SpringBootJwtApplication.class, args);

}

}

# 5. Output: