/// Controller Class

package com.cognizant.spring\_learn.Controller;  
  
import com.cognizant.spring\_learn.util.JwtUtil;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestHeader;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.web.server.ResponseStatusException;  
  
import java.util.Base64;  
import java.util.Collections;  
import java.util.Map;  
  
@RestController  
public class AuthController {  
  
 @Autowired  
 private JwtUtil jwtUtil;  
  
 @GetMapping("/authenticate")  
 public Map<String, String> authenticate(@RequestHeader("Authorization") String authHeader) {  
 String[] tokens = extractUsernamePassword(authHeader);  
 String username = tokens[0];  
 String password = tokens[1];  
 if ("Tarun".equals(username) && "Tarun2005@".equals(password)) {  
 String token = jwtUtil.*generateToken*(username);  
 return Collections.*singletonMap*("token", token);  
 } else {  
 throw new ResponseStatusException(HttpStatus.*UNAUTHORIZED*, "Invalid Credentials");  
 }  
 }  
 private String[] extractUsernamePassword(String authHeader) {  
 if (authHeader != null && authHeader.startsWith("Basic ")) {  
 String base64Credentials = authHeader.substring("Basic ".length()).trim();  
 byte[] decodedBytes = Base64.*getDecoder*().decode(base64Credentials);  
 String credentials = new String(decodedBytes);  
 return credentials.split(":", 2);  
 }  
 throw new ResponseStatusException(HttpStatus.*BAD\_REQUEST*, "Missing Authorization Header");  
 }  
}

//Config Class

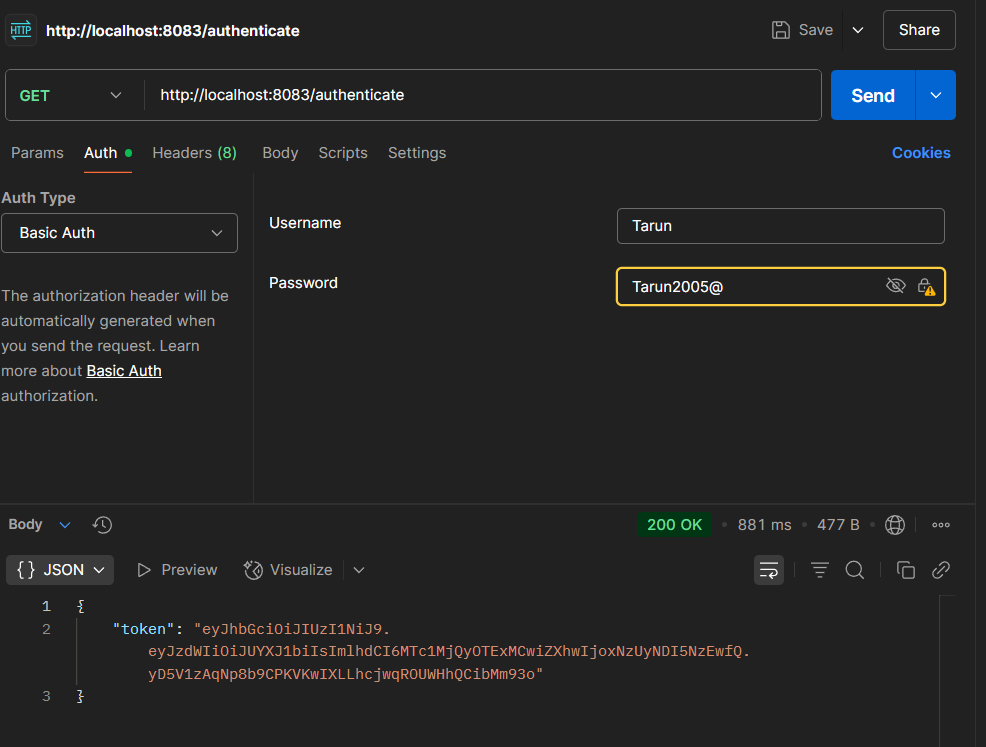
package com.cognizant.spring\_learn.config;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.Customizer;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.core.userdetails.User;  
import org.springframework.security.core.userdetails.UserDetailsService;  
import org.springframework.security.provisioning.InMemoryUserDetailsManager;  
import org.springframework.security.web.SecurityFilterChain;  
  
@Configuration  
public class SecurityConfig {  
 @Bean  
 public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {  
 return http  
 .csrf(csrf -> csrf.disable())  
 .authorizeHttpRequests(auth -> auth  
 .requestMatchers("/authenticate").permitAll()  
 .anyRequest().authenticated()  
 )  
 .httpBasic(Customizer.*withDefaults*())  
 .build();  
 }  
 @Bean  
 public UserDetailsService userDetailsService() {  
 return new InMemoryUserDetailsManager(  
 User.*withUsername*("Tarun")  
 .password("{noop}Tarun2005@")  
 .roles("USER")  
 .build()  
 );  
 }  
}

//JwtUtil Class

package com.cognizant.spring\_learn.util;  
  
import io.jsonwebtoken.\*;  
import io.jsonwebtoken.security.Keys;  
import org.springframework.stereotype.Component;  
  
import java.util.Date;  
import java.security.Key;  
@Component  
public class JwtUtil {  
 private static final Key *key* = Keys.*secretKeyFor*(SignatureAlgorithm.*HS256*);  
 private static final long *EXPIRATION\_TIME* = 600\_000;  
 public static String generateToken(String username) {  
 return Jwts.*builder*()  
 .setSubject(username)  
 .setIssuedAt(new Date())  
 .setExpiration(new Date(System.*currentTimeMillis*() + *EXPIRATION\_TIME*))  
 .signWith(*key*)  
 .compact();  
 }  
 public static String extractUsername(String token) {  
 return Jwts.*parserBuilder*().setSigningKey(*key*).build()  
 .parseClaimsJws(token)  
 .getBody()  
 .getSubject();  
 }  
 public static boolean isTokenExpired(String token) {  
 return Jwts.*parserBuilder*().setSigningKey(*key*).build()  
 .parseClaimsJws(token)  
 .getBody()  
 .getExpiration()  
 .before(new Date());  
 }  
 public static boolean validateToken(String token, String username) {  
 return *extractUsername*(token).equals(username) && !*isTokenExpired*(token);  
 }  
}

//Main Class

package com.cognizant.spring\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 }  
}

//Output