**JWT Authentication Service**

**1️ Objective**

Develop a RESTful Authentication Service using Spring Boot 3 and Spring Security, which:

* Accepts Basic Auth credentials
* Validates the user
* Generates a JWT token and returns it as JSON

**2️ Endpoints**

| Method | URL | Description |  |
| --- | --- | --- | --- |
| GET | /authenticate | Returns a JWT for valid credentials |  |

**Key Files**

**SpringlearnApplication.java**

package com.mycompany.myproject;

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);

}

}

**SecurityConfig.java**

package com.mycompany.myproject.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

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

import org.springframework.security.web.SecurityFilterChain;

import org.springframework.security.config.Customizer;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

.csrf(csrf -> csrf.disable())

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate", "/authenticate/").permitAll()

.anyRequest().authenticated()

)

.httpBasic(Customizer.withDefaults());

return http.build();

}

}

**AuthenticationController.java**

package com.mycompany.myproject.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

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

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

import jakarta.servlet.http.HttpServletRequest;

import java.util.Base64;

import java.util.Date;

@RestController

public class AuthenticationController {

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

@GetMapping("/authenticate")

public TokenResponse authenticate(HttpServletRequest request) {

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

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

throw new RuntimeException("Missing or invalid Authorization header.");

}

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

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

String credentials = new String(credDecoded);

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

String username = values[0];

String password = values[1];

if (!"admin".equals(username) || !"1234".equals(password)) {

throw new RuntimeException("Invalid Credentials");

}

String token = Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 60 \* 60 \* 1000))

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

return new TokenResponse(token);

}

public static class TokenResponse {

private String token;

public TokenResponse(String token) {

this.token = token;

}

public String getToken() {

return token;

}

}

}

**application.properties**

spring.application.name=springlearn

logging.level.com.mycompany.myproject=DEBUG

server.port=8092

**Test using Postman**

1. Method: GET
2. URL: http://localhost:8092/authenticate
3. Authorization: Basic Auth
   * Username: admin
   * Password: 1234
4. Send request → Check Headers → Ensure Authorization shows Basic YWRtaW46MTIzNA==
5. You should receive a JSON with token.

**How it works**

* /authenticate is public (permitAll()).
* The controller decodes the Basic Auth header.
* Validates the user (admin/1234).
* Generates JWT using jjwt.
* Returns JWT in JSON { "token": "<jwt>" }.