**Create authentication service that returns JWT**

**Add Dependencies**

pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-learn</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<!-- Added for JWT Authentication -->

<dependency>

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

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

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**Update Application Properties**

logging.level.com.cognizant.springlearn=DEBUG

server.port=8090

**Authentication Controller**

src/main/java/com/cognizant/springlearn/controller/AuthenticationController.java

package com.cognizant.springlearn.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

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

import com.cognizant.springlearn.service.AuthenticationService;

@RestController

public class AuthenticationController {

private static final Logger LOGGER = LoggerFactory.getLogger(AuthenticationController.class);

@Autowired

private AuthenticationService authenticationService;

@PostMapping("/authenticate")

public String authenticate() {

LOGGER.info("Starting authenticate method...");

String token = authenticationService.authenticate();

LOGGER.info("Completed authenticate method.");

return "{\"token\":\"" + token + "\"}";

}

}

No immediate output (controller and security config created)

**Authentication Service**

src/main/java/com/cognizant/springlearn/service/AuthenticationService.java

package com.cognizant.springlearn.service;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Service;

import org.springframework.web.context.request.RequestContextHolder;

import org.springframework.web.context.request.ServletRequestAttributes;

import javax.servlet.http.HttpServletRequest;

import java.util.Base64;

import java.util.Date;

@Service

public class AuthenticationService {

private static final Logger LOGGER = LoggerFactory.getLogger(AuthenticationService.class);

private static final String SECRET\_KEY = "mySecretKey1234567890"; // Should be stored securely

private static final long EXPIRATION\_TIME = 1000 \* 60 \* 60; // 1 hour

public String authenticate() {

LOGGER.info("Starting authentication process...");

// Get Authorization header

HttpServletRequest request = ((ServletRequestAttributes) RequestContextHolder.getRequestAttributes()).getRequest();

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

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

LOGGER.error("Invalid or missing Authorization header");

throw new RuntimeException("Invalid or missing Authorization header");

}

// Decode Basic Auth credentials

String base64Credentials = authHeader.substring(6);

String credentials = new String(Base64.getDecoder().decode(base64Credentials));

String[] parts = credentials.split(":");

if (parts.length != 2) {

LOGGER.error("Invalid credentials format");

throw new RuntimeException("Invalid credentials format");

}

String username = parts[0];

String password = parts[1];

// Validate credentials (hardcoded for simplicity)

if (!"user".equals(username) || !"pwd".equals(password)) {

LOGGER.error("Invalid username or password");

throw new RuntimeException("Invalid username or password");

}

// Generate JWT

String token = Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date())

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

.signWith(SignatureAlgorithm.HS256, SECRET\_KEY)

.compact();

LOGGER.info("JWT generated successfully for user: {}", username);

return token;

}

}

**Build the Project**

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

**Output**:

[INFO] Scanning for projects...

[INFO] -------------------< com.cognizant:spring-learn >-------------------

[INFO] Building spring-learn 0.0.1-SNAPSHOT

[INFO] --------------------------------[ jar ]---------------------------------

[INFO] --- maven-clean-plugin:3.2.0:clean (default-clean) @ spring-learn ---

[INFO] Deleting C:\Users\<MyUsername>\workspace\spring-learn\target

[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ spring-learn ---

[INFO] Copying 2 resources to target/classes

[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ spring-learn ---

[INFO] Changes detected - recompiling the module! :source

[INFO] Compiling 7 source files to target/classes

[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ spring-learn ---

[INFO] Copying 0 resource to target/test-classes

[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ spring-learn ---

[INFO] Compiling 1 source file to target/test-classes

[INFO] --- maven-surefire-plugin:3.0.0:test (default-test) @ spring-learn ---

[INFO] Tests are skipped.

[INFO] --- maven-jar-plugin:3.3.0:jar (default-jar) @ spring-learn ---

[INFO] Building jar: target/spring-learn-0.0.1-SNAPSHOT.jar

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 7.123 s

[INFO] Finished at: 2025-07-10T02:00:45+05:30

[INFO] ------------------------------------------------------------------------

**Test the /authenticate Endpoint in Postman**

http://localhost:8090/authenticate

 **Output**: (console output)

2025-07-10T02:02:34.123+05:30 INFO 12345 --- [http-nio-8090-exec-2] c.c.s.controller.AuthenticationController : Starting authenticate method...

2025-07-10T02:02:34.124+05:30 INFO 12345 --- [http-nio-8090-exec-2] c.c.springlearn.service.AuthenticationService : Starting authentication process...

2025-07-10T02:02:34.125+05:30 INFO 12345 --- [http-nio-8090-exec-2] c.c.springlearn.service.AuthenticationService : JWT generated successfully for user: user

2025-07-10T02:02:34.126+05:30 INFO 12345 --- [http-nio-8090-exec-2] c.c.s.controller.AuthenticationController : Completed authenticate method.