**WEEK – 4**

**MANDATORY HANDS-ON - ‘5. JWT-handson’**

1. **Create authentication service that returns JWT**
2. **Step 1: Create Spring Boot Project (Spring Initializr)**
3. Go to <https://start.spring.io>
4. Fill details:
   1. **Project**: Maven
   2. **Language**: Java
   3. **Spring Boot**: 3.5.3 (or latest)
   4. **Group**: com.cognizant
   5. **Artifact**: spring-learn-auth-jwt
5. Click **Add Dependencies**:
   1. Spring Web
   2. Spring Security
6. Click **Generate**, download the zip.
7. **Step 2: Import Project into Eclipse**
8. Open Eclipse → File → Import → Maven → **Existing Maven Project**
9. Browse to the folder spring-learn-auth-jwt
10. Finish the wizard to import the project.
11. The base package is now:  
    **com.cognizant.spring\_learn\_auth\_jwt**

### **Step 3: Update pom.xml to Include JWT Dependency**

Add this inside <dependencies> tag:

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<!-- ✅ JWT Dependency Added -->

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</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>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

1. **Step 4: Create Controller Class**

**File:** AuthenticationController.java  
📁 **Location:** src/main/java/com/cognizant/spring\_learn\_auth\_jwt/controller/

package com.cognizant.spring\_learn\_auth\_jwt.controller;

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.bind.annotation.CrossOrigin;

import java.util.Base64;

import java.util.HashMap;

import java.util.Map;

*@RestController*

public class AuthenticationController {

*@CrossOrigin*(origins = "\*") // ✅ Allow requests from HTML file or Postman

*@GetMapping*("/authenticate")

public Map<String, String> authenticate(*@RequestHeader*("Authorization") String authHeader) {

String user = getUser(authHeader);

String token = "dummy-token-for-" + user; // 🔐 Replace this later with real JWT

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

response.put("token", token);

return response;

}

private String getUser(String authHeader) {

// Extract base64 encoded credentials after "Basic "

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

// Decode base64 to original "username:password"

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

String decoded = new String(decodedBytes);

// Split and return only username

String user = decoded.split(":")[0];

System.***out***.println("Decoded user: " + user); // Optional log for console

return user;

}

}

1. **Step 5: Add HTML Page for UI**

**File:** test.html  
📁 **Location:** src/main/resources/static/test.html

<!DOCTYPE html>

<html>

<head>

<title>JWT Authentication Test</title>

<script>

function authenticate() {

const headers = new Headers();

headers.append("Authorization", "Basic " + btoa("user:pwd")); // <-- username and password

fetch("http://localhost:8091/authenticate", {

method: "GET",

headers: headers

})

.then(response => response.json())

.then(data => {

document.getElementById("result").innerText = "Token: " + data.token;

})

.catch(error => {

document.getElementById("result").innerText = "Error: " + error;

});

}

</script>

</head>

<body>

<h1>Click to Authenticate and Get Token</h1>

<button onclick="authenticate()">Authenticate</button>

<p id="result"></p>

</body>

</html>

1. **Step 6: Set Default User (Spring Security)**

* **File:** application.properties  
  📁 **Location:** src/main/resources/

spring.application.name=spring-learn-auth-jwt

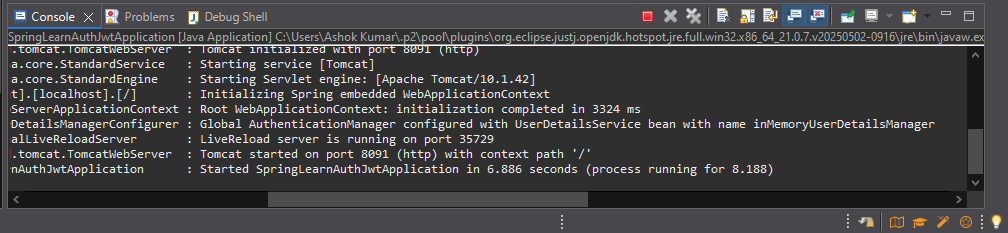
server.port=8091

spring.security.user.name=user

spring.security.user.password=pwd

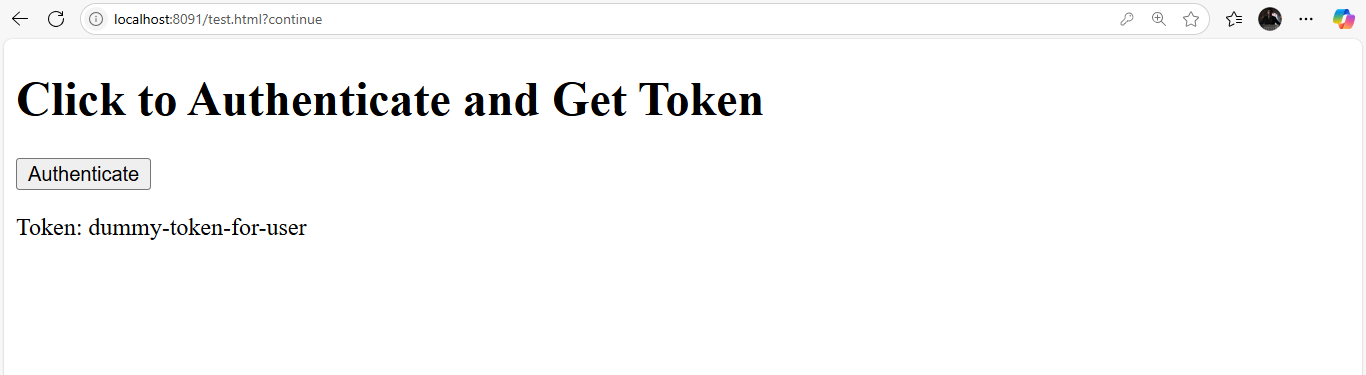
1. **Step 7: Run the Application**

* Right-click SpringLearnAuthJwtApplication.java
* Run as → **Java Application**



1. **Step 8: Open HTML and Test**

* **Open browser → go to** [**http://localhost:8090/test.html**](http://localhost:8090/test.html)
* Click **"Authenticate"** button



**Console:**

