JWT Authentication Service in Spring Boot

# 1. Introduction

This document outlines the implementation of an authentication service in a Spring Boot application that returns a JWT (JSON Web Token). It leverages Spring Security and the JJWT library for token creation and validation.

# 2. Create Authentication Service That Returns JWT

- URL: /authenticate  
- Method: GET  
- Controller: AuthenticationController  
- Functionality:  
 - Read Authorization header (Basic Auth format)  
 - Decode username and password  
 - Generate JWT token using JJWT library  
 - Return JWT token in JSON response  
  
- Sample Request:  
 curl -s -u user:pwd http://localhost:8090/authenticate  
  
- Sample Response:  
 {  
 "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2Vy..."  
 }

# 3. Key Implementation Steps

Step 1: Create Authentication Controller

- Add @RestController in package `controller`  
- Create `/authenticate` endpoint with @GetMapping  
- Read Authorization header via: @RequestHeader("Authorization") String authHeader

Step 2: Decode Username and Password

- Strip "Basic " prefix and decode using:  
 Base64.getDecoder().decode(encodedCredentials)

Step 3: Generate JWT Token

- Use JJWT library to create the token:  
 builder.setSubject(user).setIssuedAt(...).setExpiration(...).signWith(...).compact()  
- Return the token in a Map<String, String> response with key "token"

# 4. Sample cURL Testing Commands

Generate Token:  
 curl -s -u user:pwd http://localhost:8090/authenticate  
  
Access Secured Endpoint with JWT:  
 curl -s -H "Authorization: Bearer REPLACE\_TOKEN\_HERE" http://localhost:8090/countries