SmartBalaramAI: JWT Integration - Auth Service & EMI Warning Service

# 1. Overview

This document outlines the successful integration of JWT-based security between the `auth-service` and the `emi-warning-service` modules of the SmartBalaramAI platform.

Using Spring Security 6 and JWT, secure authentication and role-based authorization have been achieved across microservices.

# 2. Integration Summary

• The `auth-service` issues a JWT token after user login (based on credentials).

• The `emi-warning-service` validates the incoming JWT token on every API call using `JwtAuthFilter`.

• Role-based access (USER/ADMIN) is enforced via Spring Security configuration.

# 3. Technical Stack & Configuration

Spring Boot Version: 3.2.4

Security: Spring Security 6.x

JWT Handling: io.jsonwebtoken (jjwt) 0.11.5

Testing: JUnit 5 + Mockito + @WebMvcTest with custom mocks

Security Config: Role-based access is defined using `.requestMatchers('/api/emi/\*\*').hasRole('USER')`

# 4. Test Case Checklist

✅ 1. Register USER

✅ 2. Login USER and receive JWT

✅ 3. Access USER Endpoint - allowed

✅ 4. Access ADMIN Endpoint with USER Token - 403 Forbidden

✅ 5. Register ADMIN

✅ 6. Login ADMIN and receive JWT

✅ 7. Access ADMIN Endpoint - allowed

✅ 8. Access USER Endpoint with ADMIN Token - allowed (unless strict)

✅ 9. Access common endpoint (USER or ADMIN) - allowed

✅ 10. EMI Service token validation works with secure endpoints

✅ 11. Test cases updated and working

# 5. Key Issues Resolved

• 403 errors despite @WithMockUser → Fixed by mocking JwtService & configuring SecurityConfig.

• Missing JwtService bean in test context → Solved via @MockBean and filter exclusions.

• CSRF 403 in POST → Resolved using security configuration disabling CSRF.

• Integration test isolation between modules → Resolved using `excludeFilters` in `@WebMvcTest`.

# 6. Sample Token Flow

Login Request:

POST /api/auth/login

{ "email": "bala@mail.com", "password": "pass" }

Response:

{ "token": "<JWT\_TOKEN>" }

Secured API Call:

GET /api/emi/ping

Headers: Authorization: Bearer <JWT\_TOKEN>