



Actify-Intern-task

Tools : -

DB : Postgresql (Learn H2 in memoryDB also)

IDE : eclipse (learn how to use lombok in eclipse)

Learning Plan

1. Database Integration

- Setting up Spring Data JPA.
- Connecting to PostgreSQL/H2 DB
- CRUD operations using Spring Repository.
- Using @Entity, @Id, @GeneratedValue, and @Column.

Tasks:

- Create an Employee Management API with CRUD operations.
-

Spring Security with JWT

1. : Spring Security Basics

- Introduction to Spring Security.

- Securing endpoints with basic authentication.
- Using @EnableWebSecurity, WebSecurityConfigurerAdapter, and @Configuration.

Tasks:

- Add basic authentication to the Employee Management API.
- Create roles like ADMIN and USER.

2. JWT Authentication

- What is JWT and how it works?
- Implementing JWT in Spring Security.
- Generating and validating JWT tokens.
- Securing APIs using JWT.

Tasks:

- Create login and registration endpoints.
- Issue a JWT token upon successful login.
- Protect endpoints with JWT.

3. Role-Based Security

- Role-based access control (RBAC) in Spring Security.
- Configuring roles and permissions.
- Using @PreAuthorize, @Secured, and custom access expressions.

Tasks:

- Implement role-based access to endpoints (e.g., only ADMIN can create, USER can view).
- FreecodeCamp (shopping site backend project
<https://www.youtube.com/watch?v=oGhc5Z-WJSw&pp=ygUh3ByaW5nIHNIY3VyaXR5IDYgd2l0aCBqd3QgdG9rZW4g>)

Advanced Topics and Project Implementation

1. Validation and Exception Handling

- Using @Valid for request validation.
- Creating custom exception handlers with @ControllerAdvice.
- Global error handling.

Tasks:

- Add validation for user input in the API.
- Handle errors for invalid tokens or unauthorized access.

2. API Documentation with Swagger

- Setting up Swagger for API documentation.
- Securing Swagger endpoints.

Tasks:

- Add Swagger to your project and document all endpoints.
- Test API functionality with Swagger UI.

3. Spring Boot Testing

- Writing unit tests for services and controllers.
- Mocking with Mockito.
- Integration testing with Spring Boot Test.

Tasks:

- Write test cases for authentication and role-based endpoints.

4. Implementing Advanced Features

- Email notifications (e.g., upon registration).
- File upload and download with Spring Boot.
- Pagination and sorting for large datasets.

Tasks:

- Add email verification for user registration.
- Implement file upload/download features.

Industry-Level Project

Project 1

Create Pinacple's Project (Staykation app for property search)

Project 2

Title: Secure File Sharing System

Features:

- User authentication with JWT and role-based security (ADMIN, USER).
- File upload/download functionality.
- User registration with email verification.
- Role-based permissions (e.g., ADMIN can manage all files, USERS can manage their own).
- Swagger documentation.
- Unit and integration tests.

Project Setup

- Initialize the project.
- Set up basic user management (register, login).
- Configure JWT authentication.

Role-Based Access

- Add roles (ADMIN, USER) to the system.
- Implement role-based API access.

File Management

- Create file upload and download APIs.
- Secure file access based on user roles.

Final Touches

- Add Swagger documentation.
- Implement email notifications for file upload or registration.

Testing and Deployment

- Write unit and integration tests for critical components.
 - Deploy the project on a cloud platform (e.g., AWS, Heroku).
-

Expected Deliverables

By the end of 15 days, you should:

1. Understand the fundamentals of Spring Boot and Spring Security.
2. Have implemented JWT and role-based security.
3. Completed an **industry-level project** with features like authentication, file management, and Swagger documentation.