

6장 Spring Boot



목차

- ✓ 1. Spring Boot 개요
- ✓ 2. Thymeleaf
- ✓ 3. Lombok
- ✓ 4. 테스트와 로깅

1. Spring Boot 개요

- ✓ Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run"
- Spring Boot는 기존 Spring MVC 프로젝트를 더욱 편리하게 개발 할 수 있는 웹 애플리케이션 개발에 최적화 된 프레임워크
- Spring Boot는 복잡한 Spring 설정을 자동화하고 내장 WAS를 통해 단독 웹 애플리케이션으로 실행

강력 프레임워크

특징	Spring	Spring Boot
✓ 기본 설정	<ul style="list-style-type: none">• 기본 설정 제공하지 않음• 기본 Maven 프로젝트	<ul style="list-style-type: none">• 기본 설정 제공• Spring Initializer 제공
✓ 설정 및 구성	<ul style="list-style-type: none">• 복잡한 설정과 구성이 필요• XML 또는 Class 설정	<ul style="list-style-type: none">• 간단한 설정으로 빠르게 애플리케이션 개발• Properties 또는 YML 설정
✓ 의존성 관리	<ul style="list-style-type: none">• 개발자가 직접 의존성 관리 필요• Maven 빌드 도구	<ul style="list-style-type: none">• 자동 의존성 관리 및 스타터 패키지 제공• Gradle 빌드 도구
✓ 내장 서버	<ul style="list-style-type: none">• 내장형 서버를 제공하지 않음• war 패키지	<ul style="list-style-type: none">• 내장형 서버 제공• jar 패키지

Spring

웹 개발 (80%)

Spray Boot

JSF
↓
Spring Framework
↓
Spring Boot

2. Thymeleaf

- ✓ Thymeleaf is a modern server-side Java template engine for both web and standalone environments
- ✓ 타임리프(Thymeleaf)는 HTML 기반 템플릿 엔진으로 Spring Boot에서 JSP 대신 Thymeleaf 사용을 권장
- JSP 보다 풍부한 UI 처리 문법을 지원하여 효과적으로 동적 콘텐츠를 생성

Mustache
handlebar

특징	설명
자바 객체와의 통합	서버 사이드에서 자바 객체에 액세스하여 데이터를 템플릿에 바인딩
다양한 템플릿 속성	<ul style="list-style-type: none">HTML 요소의 다양한 속성에 데이터를 바인딩반복문, 조건문, 변수 선언 등 사용
스프링 프레임워크 통합	<ul style="list-style-type: none">스프링 프레임워크와 자연스럽게 통합되어 스프링 애플리케이션 개발에 사용스프링 부트는 기본 템플릿 엔진
자동 템플릿 캐싱	템플릿 캐싱을 통해 성능을 최적화
템플릿 재사용 및 확장	템플릿의 재사용 및 확장이 용이하며, 템플릿을 모듈화하여 관리
자바 코드와의 분리	자바 코드와 HTML 템플릿을 분리하여 디자이너와 개발자의 업무 분리

3. Lombok

- ✓ Project Lombok is a java library that automatically plugs into your editor and build tools, spicing up your java.
- Lombok은 getter, setter, toString 등의 메서드 작성 코드를 줄여주는 코드 다이어트 라이브러리
- 다양한 Annotation을 지원하며 Spring Boot 기반 프로젝트 필수 라이브러리

주요 Annotation	설명
<u>@Getter</u>	Getter 생성
<u>@Setter</u>	Setter 생성
<u>@ToString</u>	모든 필드 정보 조회 toString 생성
<u>@NoArgsConstructor</u>	기본 생성자
<u>@AllArgsConstructor</u>	모든 필드를 초기화를 위한 생성자
<u>@Data</u>	<u>@Getter + @Setter + @ToString + @RequiredArgsConstructor 통합</u>
<u>@Builder</u>	모든 필드를 초기화를 위한 Builder 패턴 메소드 생성

4. 테스트와 로깅

- TDD (Test Driven Development)는 특정 기능을 수행하는 단위 테스트를 검증하는 방식으로 테스트 코드를 반복하여 구현하는 개발 방식
- JUnit은 TDD를 수행하는 Java 단위 테스트 프레임워크
- Slf4j (Simple Logging Facade for Java)는 로깅을 위한 인터페이스를 제공하는 라이브리로 Slf4j를 구현하는 logback 라이브러리 사용

주요 Annotation	설명
@SpringBootTest	Spring Boot 애플리케이션의 통합 테스트 정의
@Test	JUnit에서 단위 테스트 메서드를 정의
@WebMvcTest	Spring MVC 통합 테스트 정의
@Before	단위 테스트 메서드가 실행되기 전에 실행할 메서드를 정의
@After	단위 테스트 메서드가 실행된 후에 실행할 메서드를 정의
@MockBean	테스트 중에 가짜(Mock) 빈을 생성하고 주입하기 위해 사용되는 어노테이션
@Slf4j	로깅을 위한 Logger 생성

@Log4j2