

# 0. 목차

---

## 1. 스프링 부트 입문

---

- Spring Framework vs Spring Boot
- Spring Boot의 철학 및 장점
- 프로젝트 생성 방법
  - Spring Initializr 사용법
  - `pom.xml` vs `build.gradle`
- 자동 설정(Auto Configuration)
- 의존성 관리 (Spring Boot Starter)

## 2. 기본 개발 환경 설정

---

- IntelliJ, VSCode 기반 개발 설정
- Maven vs Gradle 빌드 도구
- `application.properties` vs `application.yml`
- 환경 프로파일(profile) 설정
- 로그 설정 (Logback, Log4j2)
- DevTools와 자동 재시작

## 3. 핵심 아키텍처 이해

---

- IoC (제어의 역전) / DI (의존성 주입)
- Bean 생성과 생명주기
- Component Scanning
- AOP (관점 지향 프로그래밍)
- Proxy 기반 동작 원리

## 4. 웹 개발 기초

---

- Spring MVC 개념
  - DispatcherServlet 구조
  - Controller, Service, Repository 계층 설계
- `@RestController` vs `@Controller`
- 요청 매핑: `@RequestMapping`, `@GetMapping` 등
- 요청 파라미터: `@RequestParam`, `@PathVariable`, `@ModelAttribute`
- 응답 처리: `ResponseEntity`, JSON 반환

## 5. 데이터 처리 및 JPA 연동

---

- Spring Data JPA 기초
- Entity, Repository, Service 설계
- CRUD 및 페이징 처리
- JPA 쿼리 메소드 정의
- JPQL, Native Query
- 지연 로딩 vs 즉시 로딩
- N+1 문제와 해결 방법
- DTO 패턴, Projection
- QueryDSL, Jooq 개요

## 6. 예외 처리

---

- 전역 예외 처리: `@ControllerAdvice`
- 커스텀 예외 정의
- HTTP 상태 코드 매핑
- Validation & BindingResult 처리
- Bean Validation: `@Valid`, `@Validated`

## 7. 스프링 보안(Spring Security)

---

- 인증(Authentication) vs 인가(Authorization)
- Spring Security 기본 구조
- 사용자 정의 인증 필터
- JWT 기반 인증 처리
- OAuth2 로그인 처리
- CSRF, CORS, 세션 보안
- 비밀번호 암호화 (BCrypt)

## 8. 테스트 및 품질 관리

---

- 단위 테스트: JUnit 5, Mockito
- 통합 테스트: `@SpringBootTest`
- MockMvc, WebTestClient
- 테스트 슬라이스 (`@WebMvcTest`, `@DataJpaTest`)
- TestContainers 도입

## 9. 데이터베이스 연동 및 관리

---

- MySQL, PostgreSQL, Oracle 연동
- H2, SQLite 등 인메모리 DB 활용

- 트랜잭션 처리 (`@Transactional`)
- 커넥션 풀(HikariCP) 설정
- Flyway / Liquibase로 DB 마이그레이션 관리

## 10. 파일 업로드 및 다운로드

---

- MultipartFile 처리
- 대용량 파일 스트리밍
- S3, Cloud Storage 연동

## 11. RESTful API 설계

---

- REST API 원칙
- URI 설계 지침
- HATEOAS
- Swagger / Springdoc OpenAPI 연동
- Postman 테스트 자동화

## 12. 메시지 기반 비동기 처리

---

- Spring Events
- 비동기 처리: `@Async`, `@EnableAsync`
- RabbitMQ / Kafka 연동
- 메시지 리스너 및 큐 구성
- 메시지 직렬화 전략 (JSON, AVRO)

## 13. 캐싱 및 성능 개선

---

- Spring Cache (`@Cacheable`, `@CacheEvict`)
- Redis 연동
- EhCache, Caffeine 등 로컬 캐시
- 스프링 애플리케이션 성능 튜닝
- HTTP 압축, ETag 설정

## 14. 스케줄링과 작업 처리

---

- `@Scheduled`, `@EnableScheduling`
- Quartz 스케줄러 연동
- TaskExecutor, ThreadPool 설정

## 15. 외부 API 연동

---

- RestTemplate vs WebClient
- OpenFeign 사용법

- OAuth2 클라이언트 연동
- 타임아웃, 재시도, 서킷 브레이커
- Retry, Resilience4j

## 16. 모듈화 및 멀티모듈 프로젝트 구성

---

- 모노리식 구조의 한계
- 멀티 모듈 프로젝트 구성 (`:core`, `:api`, `:batch`)
- 공통 모듈, 도메인 분리 전략

## 17. 운영 및 배포

---

- JAR 배포 / WAR 배포
- Docker 이미지 빌드 및 실행
- CI/CD 파이프라인 (GitHub Actions, Jenkins)
- 무중단 배포 전략
- 프로세스 관리: systemd, supervisor, pm2

## 18. 모니터링 및 로깅

---

- Actuator 사용법
- 헬스체크, 메트릭, 트레이싱
- Prometheus + Grafana 연동
- ELK Stack(Logstash, Kibana)
- Zipkin, Jaeger 기반 분산 트레이싱

## 19. 스프링 배치(Spring Batch)

---

- Spring Batch 구조
- Job, Step, Reader, Processor, Writer
- Chunk 처리 / Tasklet 처리
- DB 연동 배치 / 파일 기반 배치
- 스케줄링 연계, 실패 복구

## 20. 스프링 클라우드 (Spring Cloud)

---

- 마이크로서비스 아키텍처 개요
- Eureka (Service Discovery)
- Config Server / Gateway / Zuul
- API Gateway 인증 통합
- Spring Cloud Bus, Sleuth, Zipkin

## 21. 실전 프로젝트 설계

---

- 도메인 주도 설계(DDD)
- TDD 기반 개발
- 헥사고날 아키텍처 적용
- CQRS / 이벤트 소싱
- MSA 전환 프로젝트 구조

## 22. 최신 기술 및 확장 주제

---

- Kotlin + Spring Boot
- GraphQL 연동
- WebSocket / STOMP
- Reactive Programming (Spring WebFlux)
- Native Image (GraalVM) 빌드