

22.12.25

빈 생명주기, 콜백, 빈 스코프

스프링 컨테이너 및 빈(싱글톤)의 생명 주기

- 스프링 컨테이너 생성 → 설정 주입 → 스프링 빈 생성 → 의존 관계 주입 → 초기화 콜백 → 사용 → 소멸 전 콜백 → 스프링 종료

콜백 메서드 지원 방식

- 인터페이스 구현 - InitializingBean, DisposableBean
- 빈 등록 초기화 메서드 - @Bean(initMethod="A", destroyMethod="B")
 - 설정 정보 사용하므로 외부 라이브러리에도 적용 가능
- 애노테이션 @PostConstruct, @PreDestroy (스프링 권장 방식)
 - 메소드에 붙여주면 끝. 단점은 외부 라이브러리에 적용 불가.

객체의 생성과 초기화는 분리하는 것이 좋다

- 생성자 → 필수 정보(파라미터)를 받아 메모리를 할당해서 객체 생성하는 책임
- 초기화 → 생성된 객체를 활용해서 외부 커넥션을 연결하는 등 무거운 동작을 수행
- 유지보수 관점에서 좋고, 초기화에 대한 지연 로딩을 줄 수 있다.

빈 스코프

- 종류
 - 싱글톤 - 스프링 컨테이너 전체
 - 프로토타입 - 스프링 컨테이너의 요청 시 생성되며, 컨테이너는 빈 생성과 의존 관계 주입 까지만 관여한다. 종료는 클라이언트가 해야 함.
 - 웹 관련
 - request - http 요청이 들어오고 나갈 때까지 유지되는 스코프(http 요청 시 빈 생성됨)

- session - 웹 세션 생성-종료 까지 유지되는 스코프
 - application - 웹의 서블릿 컨텍스트와 같은 범위로 유지되는 스코프
 - websocket
- 문제) 싱글톤 빈에 프로토타입 빈이 주입될 경우, 프로토타입 빈이 계속 유지된다.(프로토타입 빈의 주소를 받았으므로)
 - 해결책 - DL(Dependency Lookup)
 - 1) 싱글톤 빈에 필드 주입으로 ApplicationContext를 받고, 이를 통해 프로토타입 빈을 요청 - 스프링 컨테이너에 종속적인 코드 되고, 테스트 어려움
 - 2) Provider - ObjectProvider<Type> → 옵셔널, 스트림 등 편의 기능 있음. 스프링 의존적임
 - 3) Provider - Provider<Type> → JSR-330 라이브러리 사용. 자바 표준