

Spring 기본 , DL/DI

2022년 10월 31일 월요일 오후 4:17

공개

- 2003년 6월 Aparch 2.0 으로 공개
- 21년 Spring 5.0 버전까지 공개

특징

- 경량급 프레임워크
 - POJO 객체를 이용해서도 개발 가능
 - 이상한 인터페이스를 구현하거나 클래스를 상속 안받아도 됨
- 컨테이너 사용
 - 스프링 컨테이너는 Java 객체의 Life Cycle을 관리
- DI(Dependency injection) 지원
 - 설정파일 또는 어노테이션 통해서 의존관계 설정
- Aop
 - 트랜잭션, 로깅, 보안 파트 처럼 공통 요소를 핵심에서 분리하여 사용
 - 런타임시에 결합하여 진행됨
- ***Config 를 통한 PATH 설정이 아니라 -> convention over configuration 을 통한 기본 PATH설정 고정***

Bean?

- 스프링 컨테이너에 의해서 관리되는 객체
- 스프링 설정 파일에 등록되어 사용하나, 자동 등록기능 사용 가능하다.

Spring Container

- 빈이 모여 있는곳
- IoC 컨테이너로써 Context라고 표현하기도 함

DI / DL

- DL -> 저장소에 있는 객체를 찾는것
- DI -> 서로 의존성 있는 객체를 자동으로 연결해주기 (객체끼리 의존관계를 빈 설정기반(xml) 바탕하여 자동으로 연결)
 - > xml 기반이 아닌 annotation 방식으로도 가능

그럼 왜 쓰는건데?

- 매번 객체를 생성하면 매번 가비지 컬렉터 작동하고 리소스 낭비가 많음
- 스프링 컨테이너를 통해 빈 미리 생성해서 등록해두고 DL통해서 객체 찾아 DI로 연결해주는거지~
- 매번 생성하는게 아니라 불러다 쓰는 느낌쓰**