

40주차 발표_민서연

Bean

01 Spring Bean

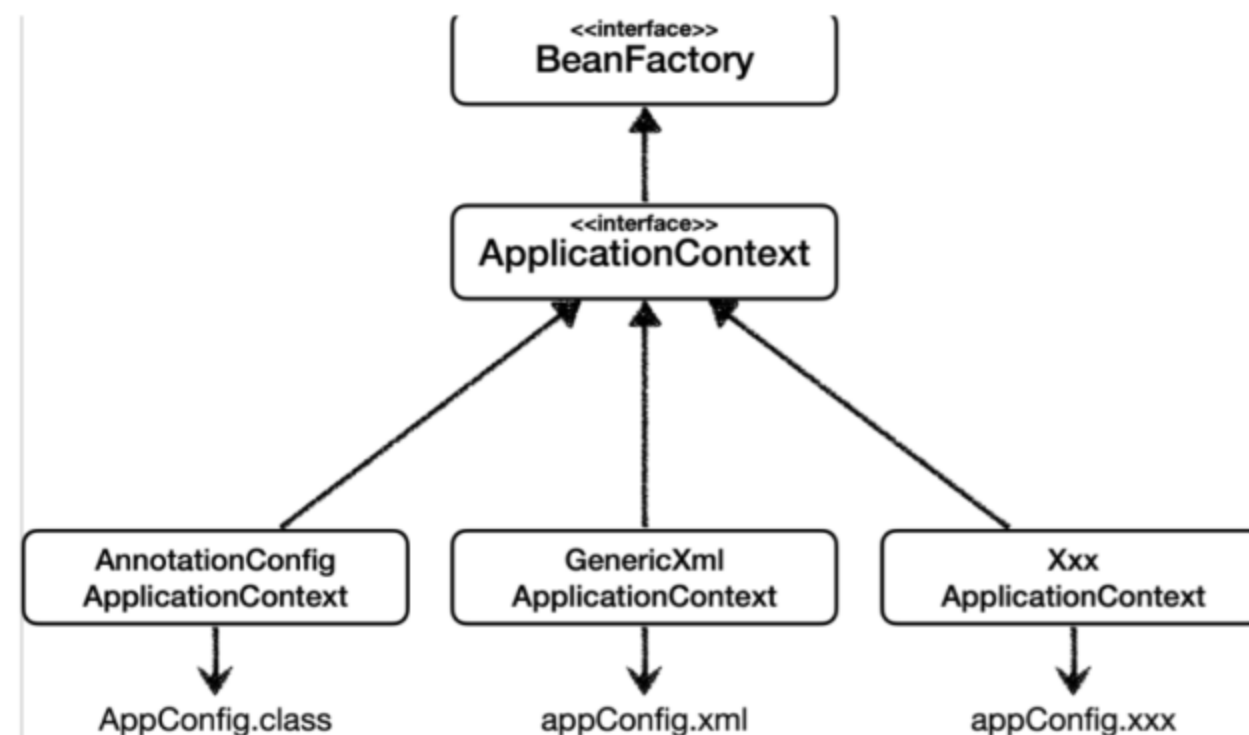
Spring Bean

스프링 컨테이너에 의해 관리되는 재사용 가능한 소프트웨어 컴포넌트

스프링 컨테이너가 관리하는 자바 객체를 의미

Spring Container

내부에 존재하는 bean의 생명 주기를 관리(생성, 관리, 제거)를 하며 생성된 bean에게 추가적인 기능을 제공한다.



01 Spring Bean

Spring Bean 사용 이유

동일한 역할을 수행하는 객체를 여러번 만들지 않아도 되기 때문

스프링 bean을 사용하지 않는다면

각각의 객체마다 new연산을 수행해 객체를 만들어야한다.

동일한 객체가 아니기 때문에 서로 가진 정보가 다르고 메모리도 비효율 적으로 사용하게 된다.

특정 클래스의 동일 인스턴스가 생길 때마다 동일한 메모리의 인스턴스를 사용하는 (싱글톤)
Spring Bean을 사용한다면 객체를 한번만 만들고도 재활용이 가능하다.

01 Spring Bean

Spring Bean 등록 방법

Component Scan

@Component를 명시해서 bean을 추가하는 방법

클래스위에 @Component를 붙이면 스프링이 알아서 스프링 컨테이너에 bean을 등록한다.

@Controller, @Service, @Repository, @Configuration은 모두 @Component의 상속을 받고 있어서 모두 Component Scan의 대상이다.

01 Spring Bean

Spring Bean 등록 방법

Java 코드로 등록

```
@Configuration
public class AppConfig {

    @Bean
    public MemberRepository memberRepository() {
        return new MemoryMemberRepository();
    }

    @Bean
    public MemberService memberService() {
        return new MemberServiceImpl(memberRepository());
    }

}
```

클래스를 생성하고 @Configuration 어노테이션을 활용한다.

bean을 등록하기 위해 인스턴스를 생성하는 메서드 위에 @Bean을 명시한다.

01 Spring Bean

Spring Bean 등록 방법

@Bean과 @Component 방식 비교

@Bean

개발자가 컨트롤이 불가능한 외부 라이브러리들을 Bean으로 등록하고 싶은 경우 사용

메소드 또는 어노테이션 단위에 붙일 수 있음

@Component

개발자가 직접 컨트롤이 가능한 클래스들의 경우에 사용

클래스 또는 인터페이스 단위에 붙일 수 있음

01 Spring Bean

Spring Bean 스코프

스코프: bean이 존재할 수 있는 범위

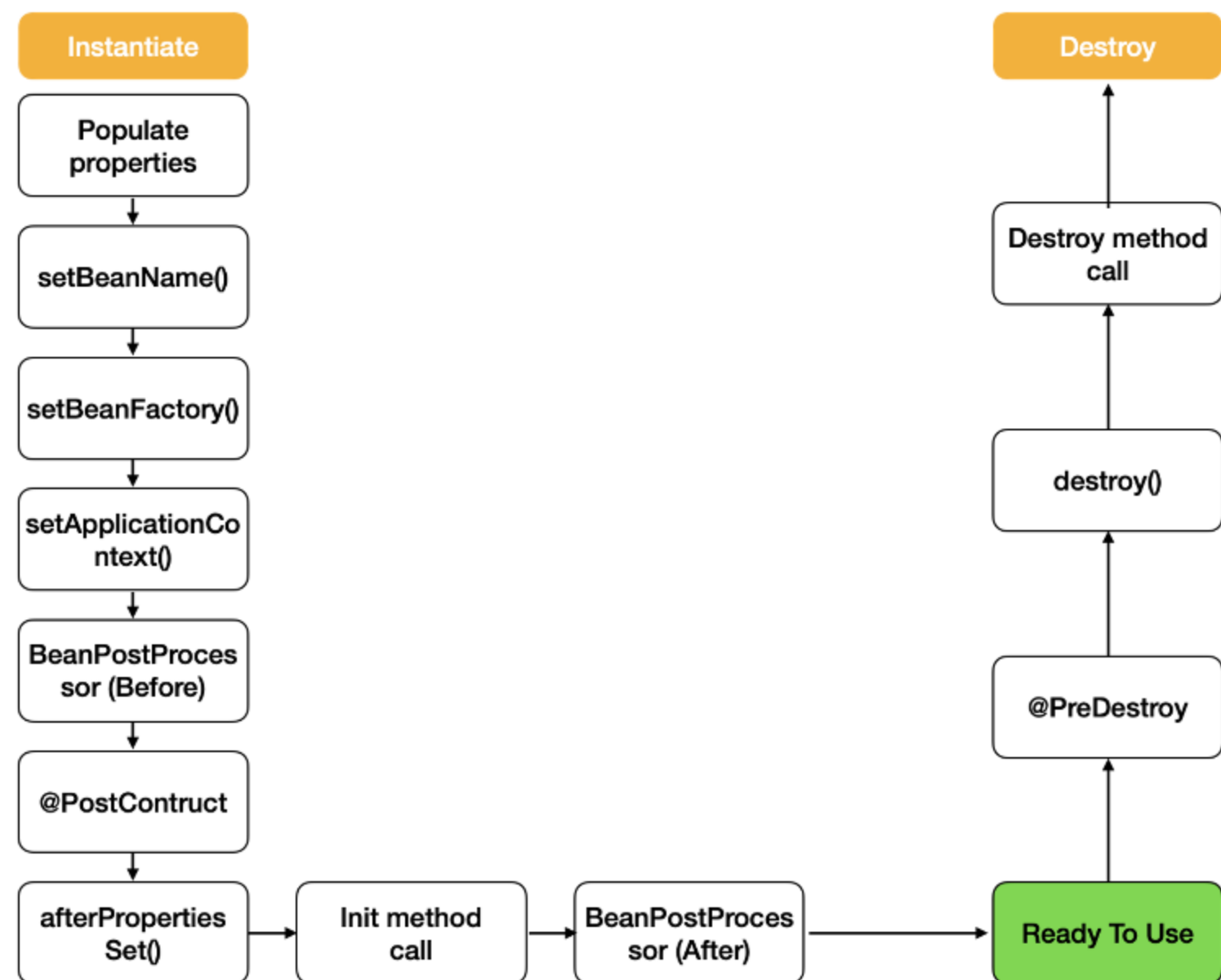
스코프 지원 방식

- 싱글톤 : 스프링 컨테이너의 시작과 종료까지 유지되는 가장 넓은 범위의 스코프
- 프로토타입: 스프링컨테이너는 프로토타입 bean의 생성과 의존 관계 주입까지만 관여하고 더는 관리하지 않는 매우 짧은 범위의 스코프
- 웹 관련 스코프
 - request: 웹 요청이 들어오고 나갈 때까지 유지되는 스코프
 - session: 웹 세션이 생성되고 종료 될때까지 유지되는 스코프
 - application: 웹의 서블릿 컨텍스트와 같은 범위로 유지되는 스코프

01 Spring Bean

Spring Bean 생명주기

Singleton Bean



스프링 컨테이너 생성

스프링 bean 생성

의존관계 주입

초기화 콜백

사용

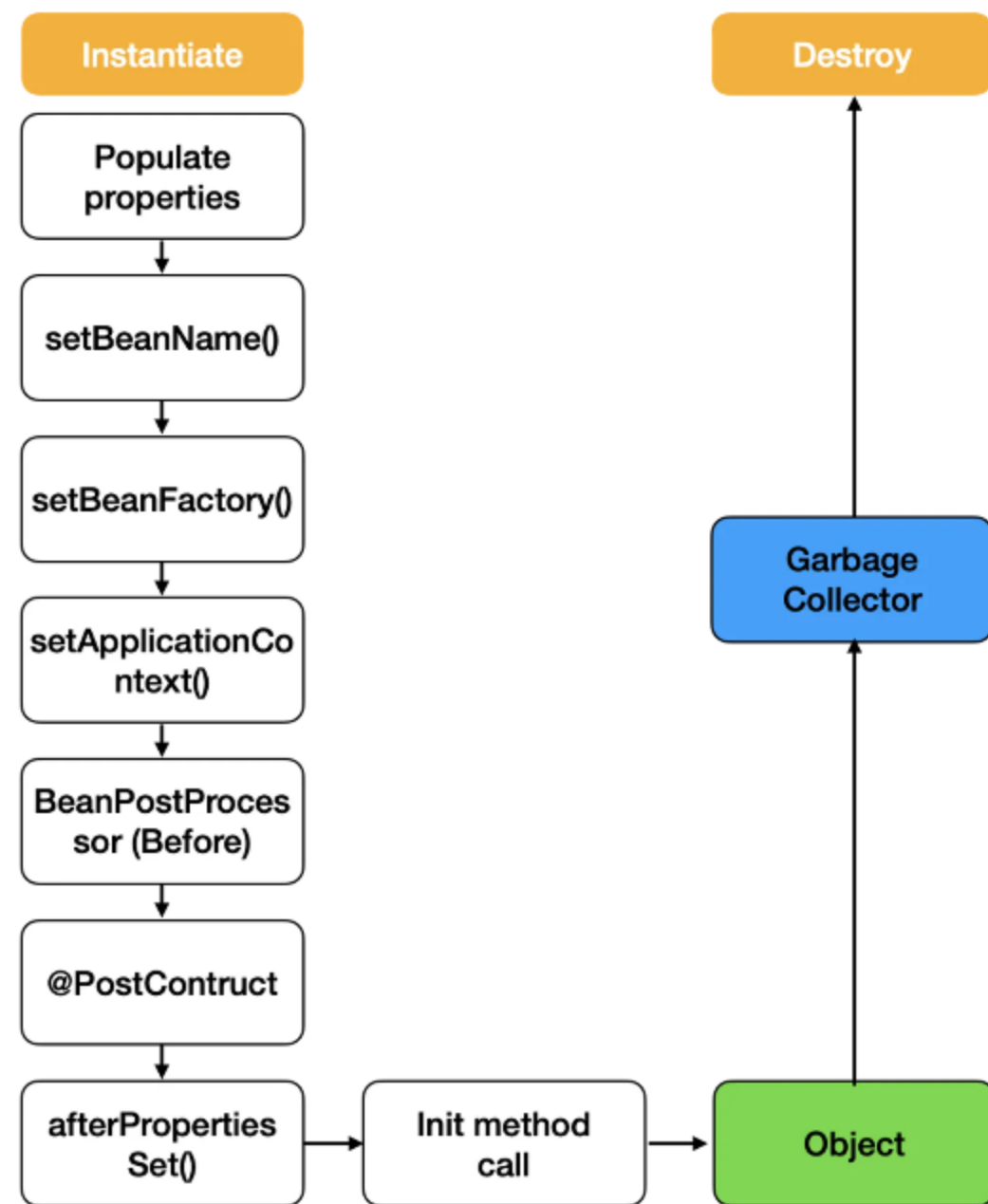
소멸 전 콜백

스프링 종료

01 Spring Bean

Spring Bean 생명주기

Prototype Bean



스프링 컨테이너 생성

스프링 bean 생성

의존관계 주입

초기화 콜백

사용

GC에 의해 수거

GC: Garbage Collector

02 출처

<https://ittrue.tistory.com/220>

<https://dev-wnstjd.tistory.com/440>

<https://velog.io/@minjiki2/Spring-%EC%99%9C-%EA%B5%B3%EC%9D%B4-spring-bean%EC%9D%84-%EC%82%AC%EC%9A%A9%ED%95%A0%EA%B9%8C>

<https://steady-coding.tistory.com/594>

<https://mslim8803.tistory.com/70>