



# 핵심원리

## 스프링 핵심원리

Aa 이름	📅 날짜	👤 담당자	📉 상태
📄 1 <u>스프링컨테이너와 스프링 빈</u>	@2021년 11월 24일 → 2021년 11월 25일	👤 DeokRin yu	🟡 진행 중

## 스프링 컨테이너

ApplicationContext를 스프링컨테이너라 한다.

기존에는 개발자가 AppConfig를 사용해서 직접 객체를 생성하고 DI를 했지만, 이제부터는 스프링 컨테이너를 통해서 사용한다.

스프링 컨테이너는 @Configuration이 붙은 AppConfig를 설정(구성) 정보로 사용한다. 여기서 @Bean 이라 적힌 메서드를 모두 호출해서 반환된 객체를 스프링 컨테이너에 등록한다. 이렇게 스프링 컨테이너에 등록된 객체를 스프링 빈이라 한다.

스프링빈은 @Bean이 붙은 메소드의 명을 스프링 빈의 이름으로 사용한다( memberService, orderService)

이전에는 개발자가 필요한 객체를 AppConfig를 사용 해서 직접 조회했지만, 이제부터는 스프링컨테이너를 통해서 필요한 스프링 빈(객체)를 찾아야 한다. 스프링 빈은 applicationContext.getBean() 메서드를 사용해서 찾을 수 있다.

기존에는 개발자가 직접 자바코드로 모든 것을 했다면 이제부터는 스프링 컨테이너에 객체를 스프링 빈으로 등록하고, 스프링 컨테이너에서 스프링 빈을 찾아서 사용하도록 변경되었다.

## 스프링 컨테이너 생성

스프링 컨테이너가 생성되는 과정을 알아보자.

```
//스프링 컨테이너 생성
ApplicationContext applicationContext = new AnnotationConfigApplicationContext(AppConfig.class);
```

- ApplicationContext를 스프링 컨테이너라 한다.
- ApplicationContext를 인터페이스이다.
- 스프링 컨테이너는 XML을 기반으로 만들 수 있고, 애노테이션 기반의 자바 설정 클래스로 만들 수 있다.
- 직전에 AppConfig를 사용했던 방식이 애노테이션 기반의 자바 설정 클래스로 스프링 컨테이너/를 만든 것이다.
- 자바 설정 클래스를 기반으로 스프링 컨테이너(ApplicationContext)를 만들어보자.
  - new AnnotationConfigApplicationContext(AppConfig.class);
  - 이 클래스는 ApplicationContext를 스프링 컨테이너라 한다.

**참고:** 더 정확히는 스프링 컨테이너를 부를 때 BeanFactory, ApplicationContext로 구분해서 이야기한다. 이 부분은 뒤에서 설명하겠다. BeanFactory를 직접 사용하는 경우는 거의 없으므로 일반적으로 ApplicationContext를 스프링 컨테이너라 한다.