

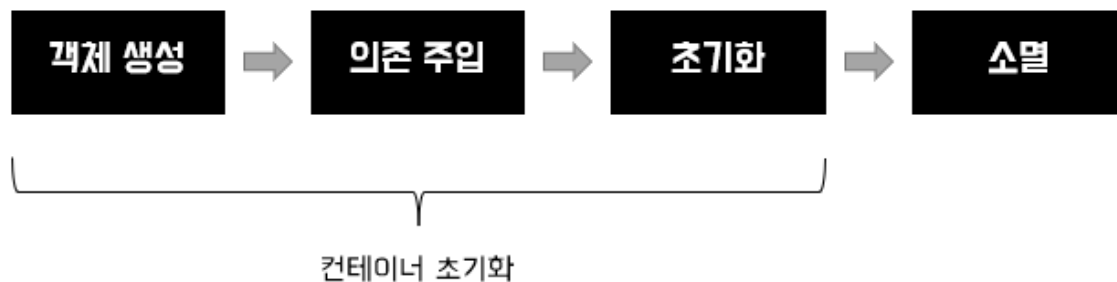


Spring Bean

스프링은 IoC (DI 컨테이너)를 통해 Bean 객체를 생성 후 관리한다.

Bean 주입 방법은 자바 XML을 이용한 방법과 어노테이션을 이용한 방법 두가지가 있으며, 스프링 부트에서는 어노테이션 기법을 주로 사용한다.

Bean Lifecycle



```
// 1. 컨테이너 초기화
ApplicationContext applicationContext =
    new AnnotationConfigApplicationContext(AppConfig.class);

// 2. 빈 사용
...

// 3. 컨테이너 종료
applicationContext.close();
```

초기화

```
public interface InitializingBean {
    void afterPropertiesSet() throws Exception;
}
```

위 인터페이스를 구현하면 스프링 컨테이너는 초기화 과정에서 빈 객체의 `afterPropertiesSet()`를 실행한다.

소멸

```
public interface DisposableBean {  
    void destroy() throws Exception;  
}
```

위 인터페이스를 구현하면 스프링 컨테이너 소멸 과정에서 `destroy()`를 실행한다.

위와 같은 초기화, 소멸 함수를 오버라이딩 하는 예시는 데이터베이스 커넥션 풀이다.

Bean 객체는 초기화 과정에서 데이터베이스 연결을 생성하고 소멸 과정에서 사용중인 데이터베이스 연결을 끊어야한다.

다른 방법

사용자가 직접 작성한 코드의 경우 `interface implement`를 할 수 있지만 그렇지 못한 경우도 분명 존재한다. 외부에서 제공받은 클래스를 스프링 빈 객체로 구현할 때 어떻게 해야할까?

```
@Bean(initMethod = "init", destroyMethod = "close")  
public void test() {  
    ...  
}
```

`@Bean` 어노테이션 속성의 `initMethod`, `destroyMethod`의 값에 적용할 함수 이름을 작성 해주면 된다.

이때 적용되는 함수에는 `Parameter`가 없어야한다. 있으면 **Exception**

이렇게 초기화를 사용할 때에는 초기화가 중복으로 일어나지 않도록 주의해줘야한다.