

Spring의 IoC와 DI

스프링이란

스프링이란, **자바 개발에 사용되는 프레임워크**이다.

프레임워크는 라이브러리처럼 개발을 빠르고 효율적이게 할 수 있는 도구이지만, 앱 흐름을 프레임워크가 주도한다는 점에서 라이브러리와 다르다. 개발자는 프레임워크가 필요한 코드를 작성해야 한다.

스프링은 (프레임워크 관점에서) 다음 3가지를 제공한다.

1. 객체의 생성 및 관리 툴 제공
2. 개발자에게 코드가 어떻게 작성되어야 하는지 기준을 제시
 - **ioc, di 에 맞춰서 객체의 생명주기와 의존관계를 개발한다**
 - **서비스 추상화**를 제공하는 것을 사용하여, 코드가 기술과 환경에 종속되지 않도록 한다
 - aop - 코드에 산재하는 부가 기능들을 모듈화한다.
3. 다양한 기술 api 제공

특히, ioc 및 di에 맞춰 개발을 하면 객체 지향의 목표 중 하나인 “유연한 확장” 설계를 도와준다.

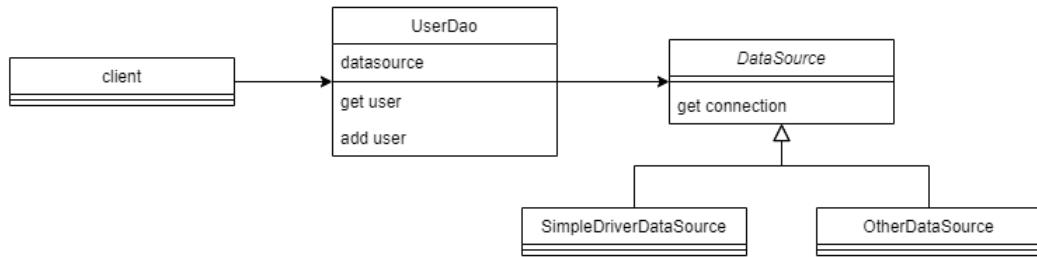
IoC, DI가 어떻게 유연한 확장을 도와주는가

IoC란

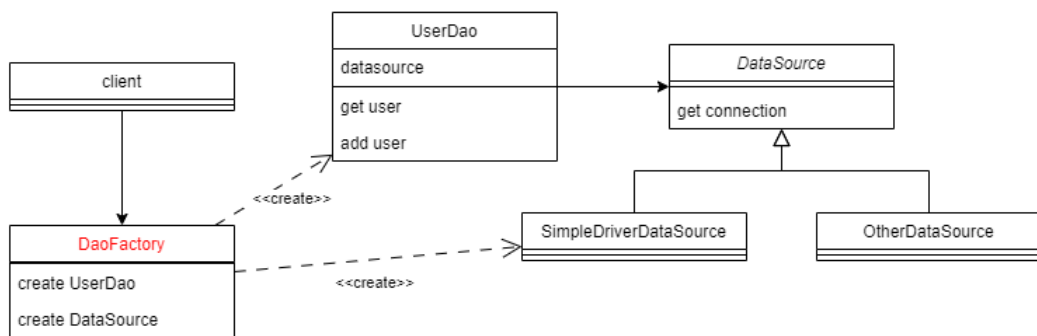
프로그램의 제어 구조가 뒤바뀌는 것이다.

즉, 객체 제어 (생성 및 관계) 권한이 제어 권한을 갖는 다른 객체(외부)에게 위임 되는 것이다.

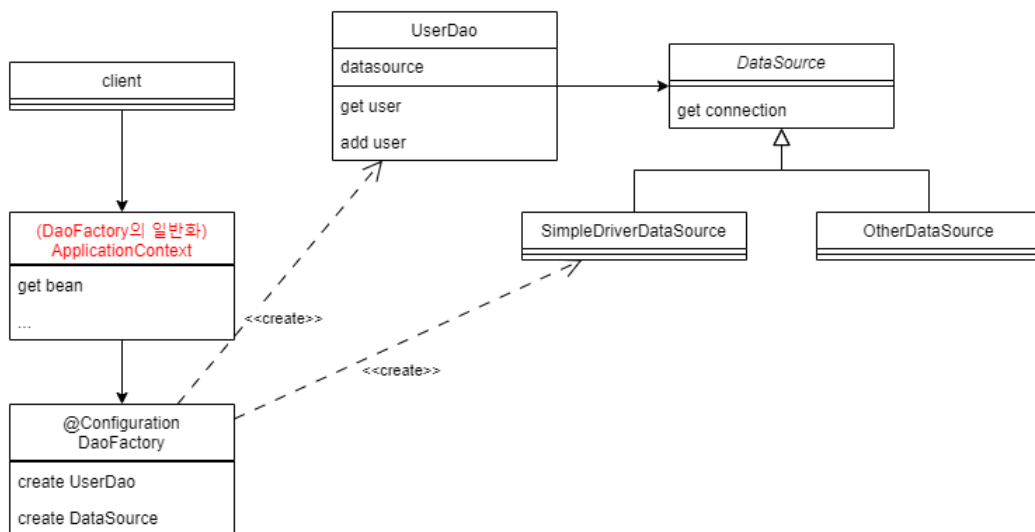
일반 코드



IoC 적용



스프링의 IoC



- 스프링에서 IoC의 역할(객체 생성 및 의존 관계 맺기)을 하는 객체는 **ApplicationContext** 이다.
- **ApplicationContext** 는 객체 제어 권한을 갖는 것 외에도 부가 기능(객체간 이벤트 발송 기능, 환경 구성 등)을 담당할 수 있는 일반화된 형태이다.
- 스프링에서는 IoC(제어의 역전)을 DI(의존성 주입)을 통해 구현하고 있다.

DI 란

D - 의존관계란

객체가 다른 객체를 사용하는 것이다. (예 : 다른 객체의 메소드 호출)

의존관계는 다음과 같은 2가지 종류가 있다.

1. 모델링 의존관계 : uml과 코드로 보이는 class 및 interface들의 연관관계
2. 런타임 의존관계 : 런타임 시에 객체들 사이에 만들어지는 의존 관계

여기서, 의존관계 주입이란, 객체를 런타임에 연결해주는 작업을 뜻한다.

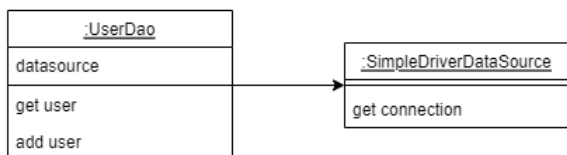
DI - 의존관계 주입이란

객체를 런타임에 연결해주는 작업이다.

다음과 같은 조건을 충족해야 한다.

1. 인터페이스에만 의존한다.
2. 런타임 시점의 의존관계를 제 3의 존재가 결정한다. (예: 팩토리, 컨테이너 등)
3. 의존관계는 객체 참조값을 외부에서 주입하면서 만들어진다.

DI - 런타임 시점



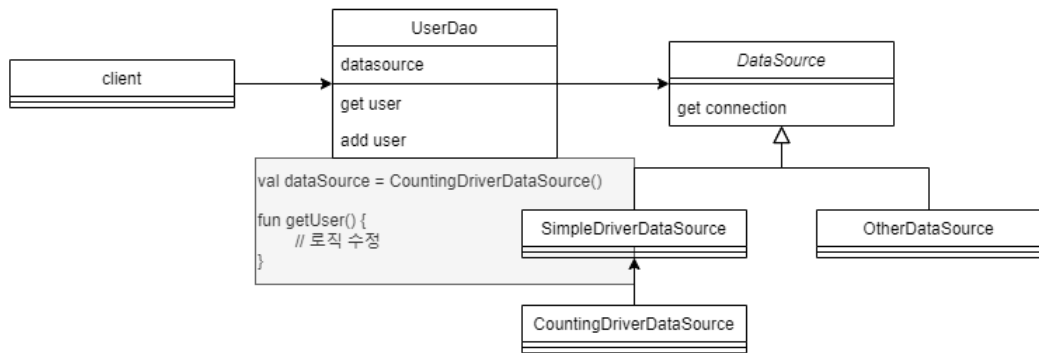
장점

1. 객체간 결합도가 낮은 코드를 제공한다.
2. 기능 변경 요청시, ocp 충족시킬 수 있다.

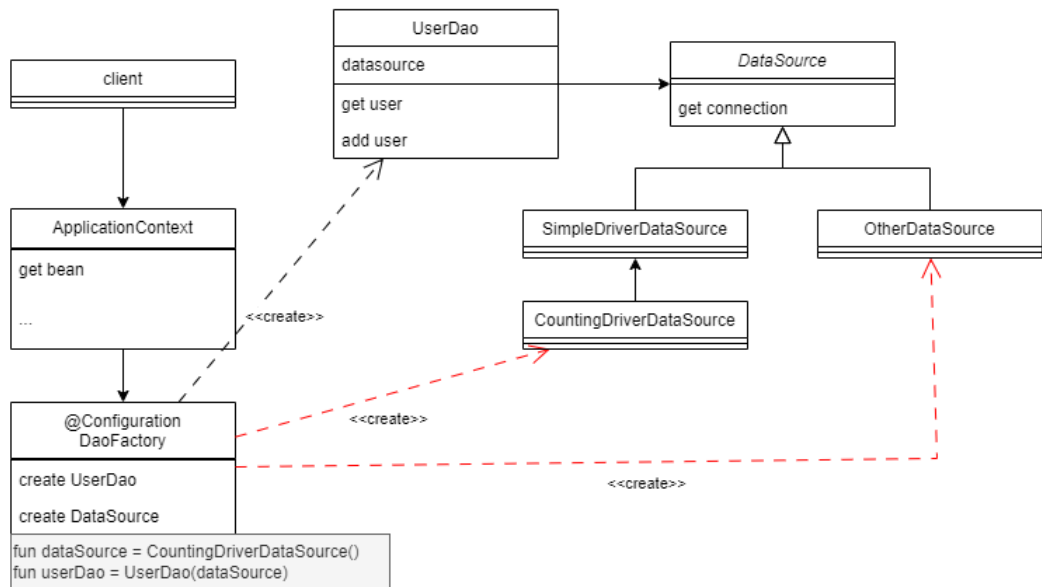
예시

- `SimpleDriverDataSource` 에서 기능 추가 요청
- `SimpleDriverDataSource` 를 대체할 기능 변경 요청

변경 요청시, 일반 코드



변경 요청시, 스프링의 IoC



(빨간줄 : 기능 확장 또는 변경 요청)

활용

- 서비스 추상화
- aop