

# Spring AOP

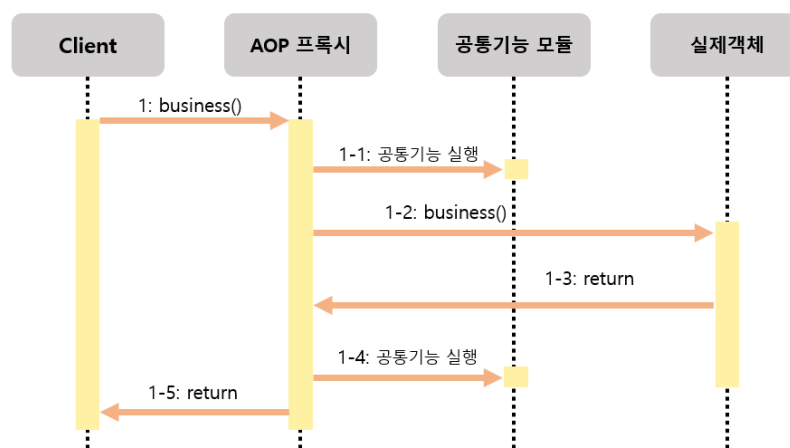
## AOP란

: AOP는 (Aspect Oriented Programming)의 약자로, 여러 객체에 공통으로 적용할 수 있는 기능을 분리해서 재사용성을 높여주는 프로그래밍 기법이다. AOP는 핵심기능과 공통기능의 구현을 분리함으로써 핵심기능을 구현한 코드의 수정 없이 공통 기능을 적용할 수 있게 만들어준다.

## AOP방식

스프링이 제공하는 AOP방식은 프록시를 이용한 방식이다.

스프링 AOP는 프록시객체를 자동으로 만들어 실제 객체의 기능을 실행하기 전.후에 공통기능을 호출한다.



AOP 주요용어

용어	의미
Advice	언제 공통기능을 핵심로직에 적용할 지를 정의 예) 메서드를 호출하기 전(언제)에 트랜잭션시작(공통 기능)기능 적용
JointPoint	Advice를 적용가능한 지점(메서드) 스프링은 프록시를 이용해서 AOP를 구현하기 때문에 메서드 호출에 대한 JointPoint만 지원
PointCut	JointPoint의 부분집합으로, 실제 Advice가 적용되는 JointPoint를 나타냄
Weaving	Advice를 핵심로직 코드에 적용하는 것
Aspect	공통기능 (클래스 기준) , Aspect = Advice(언 제)+PointCut(메소드선정알고리즘)

Advice의 종류

종류	설명
Before	대상 객체의 메서드 호출 전 공통기능 실행
After Returning	대상 객체의 메서드가 Exception없이 정상적으로 실행 된 이후 공통기능 실행
After Throwing	대상 객체의 메서드가 Exception이 발생했을 때 공통 기능 실행
After	Exception여부에 상관없이 항상실행 (finally와 비슷)
Around	메서드 실행 전,후 또는 Exception발생시점에 공통기 능 실행 다양한 시점에 원하는 기능을 삽입할 수 있기 때문에 널리사용된다.

## 스프링 AOP 구현

- Aspect로 사용할 클래스에 @Aspect애노테이션을 붙인다.
- @Pointcut애노테이션으로 공통기능을 적용한 Pointcut을 정의한다.
- 공통기능을 구현한 메서드에 원하는 시점에 따라 Advice를 택하여 애노테이션을 적용한다.