



Day6; 20220908

날짜	@2022년 9월 8일
유형	@2022년 9월 8일
태그	

spring/src at main · u8yes/spring
Contribute to u8yes/spring development by creating an account on GitHub.

u8yes/**spring**



<https://github.com/u8yes/spring/tree/main/src>

A 1 Contributor 0 Issues 1 Star 0 Forks



https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ba0f1946-69a0-40cf-b38f-57e835ebdc07/05%EC%9E%A5._Spring_AOP.pdf

API관련 JDK 각 버전 HELP

Java Platform, Standard Edition Documentation - Releases

Java Platform, Standard Edition (Java SE) helps you develop and deploy Java applications on desktops and servers. Java offers the rich user interface, performance, versatility, portability, and security that today's applications require.

🔗 <https://docs.oracle.com/en/java/javase/index.html>

<예제> 소상공인시장진흥공단_상가(상권)정보

<https://www.data.go.kr/tcs/dss/selectFileDetailView.do?publicDataPk=15083033>

pom.xml

maven repository 버전을 높이자.

```
<!-- AspectJ 제록스 회사에서 제공하는 라이브러리 -->
<dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjrt</artifactId>
    <version>${org.aspectj-version}</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver AspectJ 제록스 회사에서 제공하는 라이브러리 -->
<dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjweaver</artifactId>
    <version>1.9.7</version>
    <scope>runtime</scope>
</dependency>
```

살펴볼 어노테이션

- 어노테이션(Annotation)
 - @Aspect : Aspect 클래스 선언
 - @Before("pointcut")
 - @AfterReturning(pointcut="", returning="")
 - @AfterThrowing(pointcut="", throwing="")
 - @After("pointcut")
 - @Around("pointcut")
 - Around를 제외한 나머지 메소드들은 첫 argument로 JoinPoint를 가질 수 있다.
 - Around 메소드는 argument로 ProceedingJoinPoint를 가질 수 있다.
-

- Aspect - 여러 객체에서 공통으로 적용되는 공통 관심 사항(ex:트랜잭션, 로깅, 보안)

Pointcut을 Advice하는 설정이 Aspect

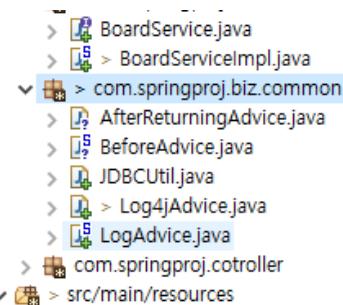
AOP 적용

```
<!-- configuration(환경설정, 구성) -->
<aop:config>
  <aop:pointcut id="implPointcut" expression="execution(* com.springproj.biz.board..*Impl.insert(..))" />
    <!-- 적용되는 메서드, 필드 // not void 리턴값이 반드시 있다, void 리턴값이 없다, * 전체를 지정,
        .. 패키지 안, 하위까지 포함대상이다., *Impl로 끝나는 글자들(클래스)이 대상이다.,
        *(..) 메서드 매개변수 타입을 제한을 두지 않겠고 모두 적용하리, (*) 메서드 안에 한 개만 받겠다.,
        *(Integer, ..) 한 개만 적용하겠고 더 와도 되고 안 와도 된다. 일단 Integer는 오길 원야함.,
        *Impl로 끝나는 클래스에서 insert... 메서드를 적용한다. -->
<aop:aspect></aop:aspect>  <!-- pointcut 설정 -->
```

AOP를 적용하고 있다는 표시



Before는 메서드 안 기능들 수행 이전에 시행, After는 메서드 후에 시행



applicationContext01.xml

```
<bean id="afterReturningAdvice" class="com.springproj.biz.common.AfterReturningAdvice"></bean>

<!-- configuration(환경설정, 구성) -->
<aop:config>
    <aop:pointcut id="getPointcut" expression="execution(* com.springproj.biz.board..*Impl.get*(..))" />

    <aop:aspect ref="afterReturningAdvice"> <!-- 설정 -->
        <aop:after-returning pointcut-ref="getPointcut" method="afterLog" returning="returnObj"/>

    </aop:aspect> <!-- pointcut 설정 -->
</aop:config>
```

AfterReturningAdvice.java

```
package com.springproj.biz.common;

import org.aspectj.lang.JoinPoint;

public class AfterReturningAdvice {

    public void afterLog(JoinPoint jp, Object returnObj) {
        String method = jp.getSignature().getName();

        System.out.println("[정상 종료] " + method + "() 비즈니스 로직 수행 후 동작");
    }
}
```

```
}
```

The screenshot shows the Eclipse IDE interface. On the left, there is a code editor with a Java configuration file named `BoardRequestClient.xml`. The file contains XML code defining beans and aspects for AOP. Lines 58 and 59 are highlighted, showing the configuration of an aspect and its after-throwing advice.

```
~
49
50
51 <bean id="afterThrowAdvice" class="com.springproj.biz.common.AfterThrowAdvice"></bean>
52
53 <!-- configuration(환경설정, 구성) -->
54<aop:config>
55   <aop:pointcut id="allPointcut" expression="execution(* com.springproj.biz.board..*Impl.*(..))" />
56
57<aop:aspect ref="afterThrowAdvice"> <!-- 설정 -->
58   <aop:after-throwing pointcut-ref="allPointcut" method="exceptionLog" throwing="exception"/>
59 </aop:aspect> <!-- pointcut 설정 -->
60
61 </aop:config>
62
63
64
```

Below the code editor is the Java Console view, which displays the following output:

```
[공통로그] insertService() 비즈니스 로직 수행 전 동작
args 정보 : BoardVO[seq=0, title=첫 게시글, writer=민용기, content=드디어 나는 개발자, date=null, cnt=0]
부적합한 값이 입력되었습니다.
[오류발생] insertService() 비즈니스 로직 수행 후 동작
Exception in thread "main" java.lang.IllegalArgumentException: 0번 글은 등록할 수 없다
at com.springproj.biz.board.service.BoardServiceTmn1.insertService(BoardServiceTmn1.java:24)
```

