Aspect oriented programming(AOP)

<https://www.journaldev.com/2583/spring-aop-example-tutorial-aspect-advice-pointcut-joinpoint-annotations>

There ways

1. Using aspectj

Step1 :- <aop:aspectj-autoproxy/>

Step2:-

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${aspectj.version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjtools</artifactId>

<version>${aspectj.version}</version>

</dependency>

Step3:- create aspect and point cut.

Step3 : define all beans in spring.xml

1. Annotation

An alternative approach is to create a custom annotation and annotate the methods where we want the advice to be applied.

1. Using configuration in spring.xml

<aop:config>

<aop:aspect ref="employeeXMLConfigAspect" id="employeeXMLConfigAspectID">

<aop:pointcut expression="execution(\* com.journaldev.spring.model.Employee.getName())" id="getNamePointcut"/>

<aop:around method="employeeAroundAdvice" pointcut-ref="getNamePointcut" arg-names="proceedingJoinPoint"/>

</aop:aspect>

</aop:config>