Spring AOP

* Aspect-Oriented Programming entails breaking down program logic into distinct parts called so-called concerns.
* The functions that span multiple points of an application are called cross-cutting concerns and these cross-cutting concerns are conceptually separate from the application's business logic.
* There are various common good examples of aspects like logging, auditing, declarative transactions, security, caching, etc.

Types of Advice

Spring aspects can work with five kinds of advice mentioned as follows −

1 – before: Run advice before a method execution.

2 – after: Run advice after the method execution, regardless of its outcome.

3 - after-returning: Run advice after a method execution only if method completes successfully.

4 - after-throwing: Run advice after a method execution only if method exits by throwing an exception.

5 – around: Run advice before and after the advised method is invoked.

AOP Terms:

Aspect

This is a module which has a set of APIs providing cross-cutting requirements. For example, a logging module would be called AOP aspect for logging. An application can have any number of aspects depending on the requirement.

Joint point

This represents a point in your application where you can plug-in the AOP aspect. You can also say, it is the actual place in the application where an action will be taken using Spring AOP framework.

Advice

This is the actual action to be taken either before or after the method execution. This is an actual piece of code that is invoked during the program execution by Spring AOP framework.

Pointcut

This is a set of one or more join points where an advice should be executed. You can specify pointcuts using expressions or patterns as we will see in our AOP examples.

Introduction

An introduction allows you to add new methods or attributes to the existing classes.

Target object

The object being advised by one or more aspects. This object will always be a proxied object, also referred to as the advised object.

Weaving

Weaving is the process of linking aspects with other application types or objects to create an advised object. This can be done at compile time, load time, or at runtime.