

# Spring 框架的 AOP

## Spring 框架的 AOP

Spring 框架的一个关键组件是**面向方面的编程(AOP)**框架。面向方面的编程需要把程序逻辑分解成不同的部分称为所谓的关注点。跨一个应用程序的多个点的功能被称为**横切关注点**，这些横切关注点在概念上独立于应用程序的业务逻辑。有各种各样的常见的很好的方面的例子，如日志记录、审计、声明式事务、安全性和缓存等。

在 OOP 中，关键单元模块度是类，而在 AOP 中单元模块度是方面。依赖注入帮助你对应用程序对象相互解耦和 AOP 可以帮助你从它们所影响的对象中对横切关注点解耦。AOP 是像编程语言的触发物，如 Perl，.NET，Java 或者其他。

Spring AOP 模块提供拦截器来拦截一个应用程序，例如，当执行一个方法时，你可以在方法执行之前或之后添加额外的功能。

## AOP 术语

在我们开始使用 AOP 工作之前，让我们熟悉一下 AOP 概念和术语。这些术语并不特定于 Spring，而是与 AOP 有关的。

项	描述
Aspect	一个模块具有一组提供横切需求的 APIs。例如，一个日志模块为了记录日志将被 AOP 方面调用。应用程序可以拥有任意数量的方面，这取决于需求。
Join point	在你的应用程序中它代表一个点，你可以在插件 AOP 方面。你也可以说，它是在实际的应用程序中，其中一个操作将使用 Spring AOP 框架。
Advice	这是实际行动之前或之后执行的方法。这是在程序执行期间通过 Spring AOP 框架实际被调用的代码。
Pointcut	这是一组一个或多个连接点，通知应该被执行。你可以使用表达式或模式指定切入点正如我们将在 AOP 的例子中看到的。
Introduction	引用允许你添加新方法或属性到现有的类中。

Target object	被一个或者多个方面所通知的对象，这个对象永远是一个被代理对象。也称为被通知对象。
Weaving	Weaving 把方面连接到其它的应用程序类型或者对象上，并创建一个被通知的对象。这些可以在编译时，类加载时和运行时完成。

## 通知的类型

Spring 方面可以使用下面提到的五种通知工作：

通知	描述
前置通知	在一个方法执行之前，执行通知。
后置通知	在一个方法执行之后，不考虑其结果，执行通知。
返回后通知	在一个方法执行之后，只有在方法成功完成时，才能执行通知。
抛出异常后通知	在一个方法执行之后，只有在方法退出抛出异常时，才能执行通知。
环绕通知	在建议方法调用之前和之后，执行通知。

## 实现自定义方面

Spring 支持 **@AspectJ annotation style** 的方法和**基于模式**的方法来实现自定义方面。这两种方法已经在下面两个子节进行了详细解释。

方法	描述
<a href="#"><u>XML Schema based</u></a>	方面是使用常规类以及基于配置的 XML 来实现的。
<a href="#"><u>@AspectJ based</u></a>	@AspectJ 引用一种声明方面的风格作为带有 Java 5 注释的常规 Java 类注释。