AOP 动态代理

关注点：增加某个业务：如安全、缓存、异常等。  
**切面（Aspect）**：一个关注点的模块化。  
  **举个例子，日志一般都会在每个方法中都要添加，可以用一个方法将日志封装成一个类。这也就是模块化的理念**。

其本质——将公共的业务（日志、安全等）和领域业务结合。当执行领域业务时就会把公共业务加进来，实现公共业务的重复利用。使得领域业务更加纯粹。

### JDK动态代理和CGLIB代理生成的区别

JDK动态代理只能对实现了接口的类生成代理，而不能针对类 。  
CGLIB是针对类实现代理，主要是对指定的类生成一个子类，覆盖其中的方法 。

(1)当Bean实现接口时，Spring就会用JDK的动态代理

(2)当Bean没有实现接口时，Spring使用CGlib是实现

**三、CGlib比JDK快？**

　 (1)使用CGLib实现动态代理，**CGLib底层采用ASM字节码生成框架，使用字节码技术生成代理类，比使用Java反射效率要高。唯一需要注意的是，CGLib不能对声明为final的方法进行代**理，因为CGLib原理是动态生成被代理类的子类。 (2)在对JDK动态代理与CGlib动态代理的代码实验中看，1W次执行下，JDK7及8的动态代理性能比CGlib要好20%左右。

二是采用静态织入的方式，引入特定的语法创建“方面”，从而使得编译器可以在编译期间织入有关“方面”的代码。

Spring提供了两种方式来生成代理对象: JDKProxy和Cglib，具体使用哪种方式生成由**AopProxyFactory根据AdvisedSupport对象的配置来决定。**

**默认的策略是如果目标类是接口，则使用JDK动态代理技术，如果目标对象没有实现接口，则默认会采用CGLIB代理。**

如果目标对象实现了接口，可以强制使用CGLIB实现代理（添加CGLIB库，**并在spring配置中加入<aop:aspectj-autoproxy proxy-target-class="true"/>**）

Spring AOP部分使用JDK动态代理或者CGLIB来为目标对象创建代理。（建议尽量使用JDK的动态代理）

如果被代理的目标对象实现了至少一个接口，则会使用JDK动态代理。所有该目标类型实现的接口都将被代理。若该目标对象没有实现任何接口，则创建一个CGLIB代理。因为JDK动态代理不依赖其他包，Cglib需要导入ASM包，对于简单的有接口的代理使用JDK动态代理可以少导入一个包。

pring有两种动态代理方式，一种就是[jdk](https://www.baidu.com/s?wd=jdk&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rADsnHDdnWbsPWnvnvf40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWcYrjfdnH03PWczn1cznHmY" \t "https://zhidao.baidu.com/question/_blank)动态代理，还有就是cglib动态代理，[jdk](https://www.baidu.com/s?wd=jdk&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rADsnHDdnWbsPWnvnvf40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWcYrjfdnH03PWczn1cznHmY" \t "https://zhidao.baidu.com/question/_blank)动态代理只能代理接口，cglib可以是类。**[jdk](https://www.baidu.com/s?wd=jdk&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rADsnHDdnWbsPWnvnvf40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWcYrjfdnH03PWczn1cznHmY" \t "https://zhidao.baidu.com/question/_blank)代理生成速度比cglib快，性能没有cglib好**

。