

Spring相关教程/资料

官网相关

- [Spring官网](#)、[Spring系列主要项目](#)、[Spring官网指南](#)、[官方文档](#)
- [spring-framework-reference](#)
- [Spring Framework 4.3.17.RELEASE API](#)

系统学习教程

文档

- [极客学院Spring Wiki](#)
- [Spring W3Cschool教程](#)

视频

- [网易云课堂——58集精通java教程Spring框架开发](#)
- [慕课网相关视频](#)
- **黑马视频和尚硅谷视频（非常推荐）：** 微信公众号：“**JavaGuide**”后台回复关键字“**1**”免费领取。

面试必备知识点

SpringAOP,IOC实现原理

AOP实现原理、动态代理和静态代理、Spring IOC的初始化过程、IOC原理、自己实现怎么实现一个IOC容器？这些东西都是经常会被问到的。

推荐阅读：

- [自己动手实现的 Spring IOC 和 AOP - 上篇](#)
- [自己动手实现的 Spring IOC 和 AOP - 下篇](#)

AOP

AOP思想的实现一般都是基于 **代理模式**，在JAVA中一般采用JDK动态代理模式，但是我们都知**道，JDK动态代理模式只能代理接口而不能代理类**。因此，Spring AOP 会这样子来进行切换，因为Spring AOP 同时支持CGLIB、ASPECTJ、JDK动态代理。

- 如果目标对象的实现类实现了接口，Spring AOP 将会采用 JDK 动态代理来生成 AOP 代理类；
- 如果目标对象的实现类没有实现接口，Spring AOP 将会采用 CGLIB 来生成 AOP 代理类——不过这个选择过程对开发者完全透明、开发者也无需关心。

推荐阅读：

- [静态代理、JDK动态代理、CGLIB动态代理讲解](#)：我们知道AOP思想的实现一般都是基于 **代理模式**，所以在看下面的文章之前建议先了解一下静态代理以及JDK动态代理、CGLIB动态代理的实现方式。

- [Spring AOP 入门](#)：带你入门的一篇文章。这篇文章主要介绍了AOP中的基本概念：5种类型的通知（Before, After, After-returning, After-throwing, Around）；Spring中对AOP的支持：AOP思想的实现一般都是基于代理模式，在Java中一般采用JDK动态代理模式，Spring AOP 同时支持 CGLIB、ASPECTJ、JDK动态代理，
- [Spring AOP 基于AspectJ注解如何实现AOP](#)：**AspectJ是一个AOP框架，它能够对java代码进行AOP编译（一般在编译期进行），让java代码具有AspectJ的AOP功能（当然需要特殊的编译器），可以这样说AspectJ是目前实现AOP框架中最成熟，功能最丰富的语言，更幸运的是，AspectJ与java程序完全兼容，几乎是无缝关联，因此对于有java编程基础的工程师，上手和使用都非常容易。Spring注意到AspectJ在AOP的实现方式上依赖于特殊编译器(ajc编译器)，因此Spring很机智回避了这点，转向采用动态代理技术的实现原理来构建Spring AOP的内部机制（动态织入），这是与AspectJ（静态织入）最根本的区别。Spring 只是使用了与 AspectJ 5 一样的注解，但仍然没有使用 AspectJ 的编译器，底层依是动态代理技术的实现，因此并不依赖于 AspectJ 的编译器。**Spring AOP虽然是使用了那一套注解，其实实现AOP的底层是使用了动态代理(JDK或者CGLib)来动态植入。至于AspectJ的静态植入，不是本文重点，所以只提一提。
- [探秘Spring AOP（慕课网视频，很不错）](#)：慕课网视频，讲解的很不错，详细且深入
- [spring源码剖析（六）AOP实现原理剖析](#)：通过源码分析Spring AOP的原理

IOC

- [\[Spring框架\]Spring IOC的原理及详解。](#)
- [Spring IOC核心源码学习](#)：比较简短，推荐阅读。
- [Spring IOC 容器源码分析](#)：强烈推荐，内容详尽，而且便于阅读。
- [Bean初始化过程](#)

Spring事务管理

- [可能是最漂亮的Spring事务管理详解](#)
- [Spring编程式和声明式事务实例讲解](#)

Spring单例与线程安全

- [Spring框架中的单例模式（源码解读）](#)：单例模式是一种常用的软件设计模式。通过单例模式可以保证系统中一个类只有一个实例。spring依赖注入时，使用了 多重判断加锁 的单例模式。

Spring源码阅读

阅读源码不仅可以加深我们对Spring设计思想的理解，提高自己的编码水平，还可以让自己在面试中如鱼得水。下面的是Github上的一个开源的Spring源码阅读，大家有时间可以看一下，当然你如果有时间也可以自己慢慢研究源码。

- [spring-core](#)
- [spring-aop](#)
- [spring-context](#)
- [spring-task](#)
- [spring-transaction](#)
- [spring-mvc](#)
- [guava-cache](#)