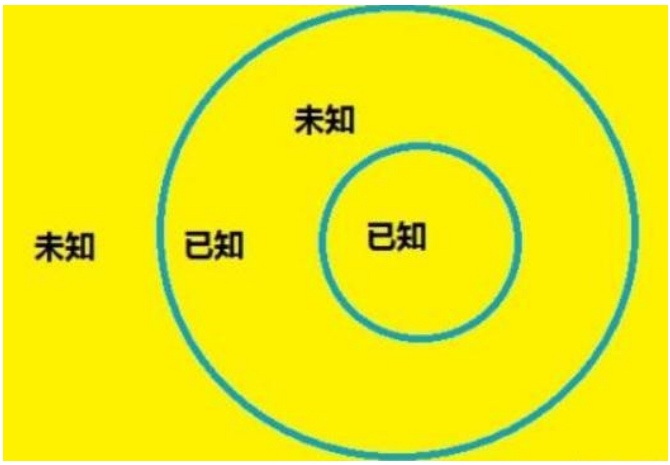
02 为什么要深入Spring5的核心原理或者源码？

更新时间：2020-05-28 11:33:42

> 世上无难事,只要肯登攀。——毛泽东

背景

当程序员当的越久，接触的越多，就会越“迷茫”。Java 好像懂，Spring、Hibernate、Mybatis 也好像懂一点，Mysql、Redis、Mongo、ES/Solr 也会用。技术栈越来越多，但是好像都懂一点，但哪一点也不精通，碰到跳槽面试被追根究底的问，就 Game Over 了。



为什么选择 Spring Framework？

那么怎么培养或者锻练出超出同龄人的独门利器呢？很多人没有方向，个人推荐从 Spring 开始，其中一个原因是 Spring 串起来了上面所有的知识，还有很多延展，通过学习 Spring 就可以把它们都学会。**按照 Spring 官方最新的说法，Spring 是现代 Java 的起源。** 在 Java 生态系统中，Spring 已经是占据了无可撼动的地位。随着微服务等概念的越来越火，Spring Boot 等也是越来越受欢迎了。主流的框架已经从 SSH 演化成 Spring 全家桶。

谁也无法否认，Spring 无论在 Java 生态系统，还是在就业市场，是绝对的王者。面试出镜率之高，投产规模之广，无出其右。随着技术的发展，Spring 从往日的 IoC 框架，已发展成 Cloud Native 基础设施，衍生出大量 Spring 技术栈，如大家熟知的 Spring Boot、Spring Cloud 和 Spring Security 等。因此，Spring 生态体系随之变得庞大，如 Apache Dubbo 对 Spring Framework 的整合，各类中间件 Spring Boot Starter，以及多种 Spring Cloud 实现方案。

作为 Spring 技术生态的基石，Spring Framework 无论在设计，还是在实现上，都是一个优秀的框架，许多细节方面是非常值得我们学习的，当然，作为面向钱途编程社群的一份子，我们不但要关注技术面的发展，而且也要重视经济面的收益。**深入掌握 Spring Framework，无疑是你进军大厂、获得更好的职业发展必须跨过的一道难关。**Spring 具备巨大的优势，包括 API 抽象硬核实力，模块化设计、功能的稳定性、可扩展性和可测试性。

由于 Spring 所整合的 Java 生态是完整的，也是庞大的。在时间拮据的前提下，面对浩如烟海的技术体系，需要有人来指点迷津，需要踩坑经验来弯道超车，需要实践来巩固效果 。

针对 Spring 的源码做扩展也比较复杂。Springframework 项目中提供的组件比较丰富，每个组件都有其对应的功能，能不能合理利用起来 Spring 提供的组件是考察一个程序员对 Spring 应用熟悉程度的重要标准。Spring 的源码设计的比较优秀，利用了很多优秀的设计模式，需要考虑如何把这种设计模式利用到自己写的代码中需要好好借鉴、学习 Spring 源码中作者的思想。

我是这么读源码的

学好 Spring，对程序员找工作、面试有非常大的帮助。比如最新的 Spring5 的新特性很多人都不理解，假设程序员能够读一遍 Spring 源码把 Spring 的设计思想完全理解，那么可以秒杀很多面试官和程序员。并且可以利用 Spring 提供的扩展写出很多优秀的代码甚至中间件。

1. 读源码的经历

刚参加工作那会，没想过去读源码，更没想过去改框架的源码；总想着别人的框架应该是完美的、万能的，应该不需要改；另外即使我改了源码，怎么样让我的改动生效了？项目中引用的不还是没改的 jar 包吗。回想起来觉得那时候的想法确实挺……

工作了一两年后准备跳槽，开始了一轮的面试，其中有几个面试官就问到了相关的源码问题：ArrayList、HashMap 的底层实现，Spring、Mybatis 的相关源码。问源码的面试一般就是回去等消息，然后就没然后了。

2. 我为什么读源码

很多人一定和我一样的感受：源码在工作中有用吗？用处大吗？很长一段时间内我也有这样的疑问，认为那些有事没事扯源码的人就是在装，只是为了提高他们的逼格而已。

那为什么我还要读源码呢？刚开始为了面试，后来为了解决工作中的问题，再后来就是个人喜好了。说的好听点是有匠人精神；说的委婉点是好奇（底层是怎么实现的）；说的不自信点是对黑盒的东西我用的没底，怕用错；说的简单直白点是提升自我价值，为了更高的薪资待遇（这里对真正的技术迷说声抱歉）。

源码中我们可以学到很多东西，学习别人高效的代码书写、学习别人对设计模式的熟练使用、学习别人对整个架构的布局，等等。如果你还能找出其中的不足，那么恭喜你，你要飞升了！会使用固然重要，但知道为什么这么使用同样重要。从模仿中学习，从模仿中创新。

读源码不像围城（外面的人想进来，里面的人想出去），它是外面的人不想进来，里面的人不想出去；当我们跨进城内，你会发现（还是城外好，皮！）城内风光无限，源码的海洋任我们遨游！

3. 我是怎么样读源码的

通过合适的例子配合 IDE 进行断点追踪

面对未知的、茫茫多的源码，我们往往没有足够的时间、经历和耐心去通读所有源码，我们只需要去读我们关注的部分即可（有人可能会说我都不关心，这…）。合适的例子配合源码的阅读才是读源码的利器，这方面老程序员有更深刻的体会。

官方用户文档，亲生父母往往对孩子是最了解的，对孩子的描述也是最详细的；比如 Spring-framework-reference 就是对 Spring 最详细的描述，怎么样使用 Spring、Spring 特性等等，通过此指南，Spring 在你面前一览无遗；

但是，Spring 毕竟是外国人的孩子，如果英语不好，估计读起来有点头疼了，不过我们有 Google 翻译呀，咬咬牙也是能看的。源码世界的丈母娘、老岳丈是非常慷慨的！

其次是书籍，国外优秀的有很多，国内也不乏好书，比较推荐此方式，自成体系，让我们掌握的知识点不至于太散。这就是好比是源码的闺蜜，对源码非常了解，重点是挺大方，会尽全力帮助我们了解源码。

最后就是博客，虽然可能觉得知识点比较散，但是针对某个知识点却特别的细，对彻底掌握非常有帮助，网上就有很多技术大牛，写的博客自然也是非常棒，非常具有学习价值。当然还有社区、论坛、Github、码云等等。这就是源码的朋友圈，我们从中也能获取到非常多关于源码的信息。

}

01 开篇：在Spring Boot，Spring Cloud大行其道的今天Spring 过时了吗？

03 深入Spring5 的原理或者源码的道和术