

不使用Spring的5个理由 PDF转换可能丢失图片或格式，建议  
阅读原文

[https://www.100test.com/kao\\_ti2020/240/2021\\_2022\\_\\_E4\\_B8\\_8D\\_E4\\_BD\\_BF\\_E7\\_94\\_A8S\\_c104\\_240573.htm](https://www.100test.com/kao_ti2020/240/2021_2022__E4_B8_8D_E4_BD_BF_E7_94_A8S_c104_240573.htm) 我不愿使用Spring有几个理由：1. Spring的配置臃肿 我的项目组在开发一个企业级应用时，使用了依赖注入框架。这个项目中，有1500多个类，并且分散在超过11个的模块里。以我在实际开发中的经验，我们创建出的service对象应该少于依赖他们的其他对象。如果我们使用了Spring框架，当我们创建需要依赖100个service对象的1000个action对象时，这就意味着我们要对这1000个bean做配置工作。如果action的数量还在不断增加，这项工作将变得更加糟糕。我们试图重构一些东西、而又不愿破坏已有的代码，就必须加倍小心。你或许想到了通过类型（byType）来自动绑定，哦？这或许不是一个坏主意。可是，为什么不通过名称（byName）来自动绑定呢？可是如果我们对不同的对象做配置就有不同的名称，这听上去很容易让人糊涂，那样的话，我猜你又得在办公室里度过漫漫长夜了。2. XML文件配置痛苦 XML配置痛苦，这个痛苦不是说编写它有多复杂，更多是指其维护性。如果你有1000个action，你需要对在配置中放置什么和如何放置很清楚，你需要有只鹰般锐利的眼睛，你必须不能忘记在改动XML配置时使用工具来查找和替换，否则，这个应用程序会在产品化的时候崩溃。3. 如果使用XML配置，你将弱化Java强类型检查 当你开始使用XML配置的时候，你将弱化Java的强大。当你幸运地发现注入到bean里的这个对象不是这个bean所需要的，但你必须等待下去直到Spring容器开始启动并且检查依赖关系。在这个时候

，你该意识到你犯了个愚蠢的错误。哎！一些配置不使用XML，而使用Java类，在Guice里，你可以使用module。如果我们想要灵活性，我们仍然可以通过分离业务逻辑包到另外的包中来达到这点，并且在核心包中，你只需使用Class.forName（“ the module class ”）。这就是全部所在！

4. Spring不是轻量级的容器 不幸地是，Spring不再是轻量级容器。现在，Spring的性能不再是最快的了，已经有很多性能更好的轻量级容器出现了。

5. Spring是一个希望我们构建松耦合程序的容器 Spring是一个只是希望我们使用松耦合技术的容器，Spring没有真正地更多关注紧耦合。我非常确定，一旦我们使用除了spring-core.jar的Spring包，这将意味着我们的程序不能离开Spring存活。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)