

# 开篇词：为什么要学习 Spring Boot

---

大家好，我是纯洁的微笑，从 2013 年到现在，我一直从业在第三方支付和互联网金融这两个领域。在互联网金融公司工作期间，从零参与公司技术平台建设，随着公司业务不断发展，组织技术团队对平台进行过四次大架构升级，从最初单体架构发展到最后的微服务架构，我也从一个一线的开发人员成长为互联网金融公司的技术负责人。

2016 年，也就是构建第四代平台架构的时候，技术调研时发现到了 Spring Boot，初步尝试使用后，被 Spring Boot 简洁快速开发的魅力所吸引，随即推广到公司所有项目中。为了促进学习，我在网络上连载了 Spring Boot 的系列文章，意外得到广大网友的认可。早期的文章可以在我的个人公众号或者博客查看：

- [我的公众号](#)
- [我的博客](#)

2017 年我又从互联网金融公司回归到第三方支付行业，当时目前的公司正在构建新一代的支付平台，技术栈和互金的第四代平台一样，采用的 Spring Boot + Spring Cloud 技术栈，有所不同的是支付行业所面临的业务更加复杂，交易规模更为庞大。在互联网金融公司，第四代架构大概有二十多个微服务支撑整个平台，现在公司的新一代支付系统微服务项目六十多个，每个服务部署三个实例就高达一百多个；在互金行业只有在抢标的时候有大量的流量，一年交易二、三十亿，现在我就职的支付公司每天交易量就高达四、五十亿，对系统的稳定性和性能要求更高。

回到第三方支付行业后，开始负责公司新一代平台的微服务架构实践，以及微服务架构下大数据平台建设。经过两家公司对 Spring Boot 的使用，越发认识到 Spring Boot 对传统开发带来的挑战，Spring Cloud 未来可能会随着 Kubernetes 和 Service Mesh 的发展有所弱化，但是 Spring Boot 给开发带来的影响是深远的。恰逢今年 Spring Boot 2.0 的推出，可以预见的是未来 Spring Boot 的发展会越来越好。

## Spring Boot 2.0 课程

Spring Boot 2.0 的推出又激起了一阵学习 Spring Boot 热，单从我个人博客访问量大幅增加，就可以感受到大家对学习 Spring Boot 的热情。去年我在 GitChat 平台上发布的达人课 [《快速学习 Spring Boot 技术栈》](#) 销量达 2400 多份，现在仍然还在不断的增长中，说明行业内对学习 Spring Boot 的需求在不断地上涨。

去年在写 Spring Boot 课程时版本是 1.5.8，在今年的 3 月 Spring 官方发布了 Spring Boot 2.0，2.0 在 1.0 的基础上进行了大幅优化，2.0 集成了很多最新优秀的技术和新特性，对性能的提升非常的明显。

Spring Boot 2.0 升级的同时在 1.0 的基础上大量优化了 API 的使用，很多 1.0 的 API 在 2.0 版本中已经不适用，在 2.0 版本中使用之前的 API 操作往往会报错，我也收到了大量类似的留言。结合这些反馈信息，本课程会参考以上内容的基础上完全重新设计，课程的变动点如下：

- 2.0 的课程将包含 1.0 的课程，并且会对 1.0 的课程内容进行升级，Spring Boot 使用的版本升级为 2.X；
- 1.0 只是将当时最常用的实战功能进行了提炼，2.0 课程将会系统性介绍 Spring Boot 各种使用场景。

- 课程内容和技術棧都會使用最新穩定版本，課程數量也由以前的 16 課變為現在的 42 課，相當於寫了一本 Spring Boot 小書。

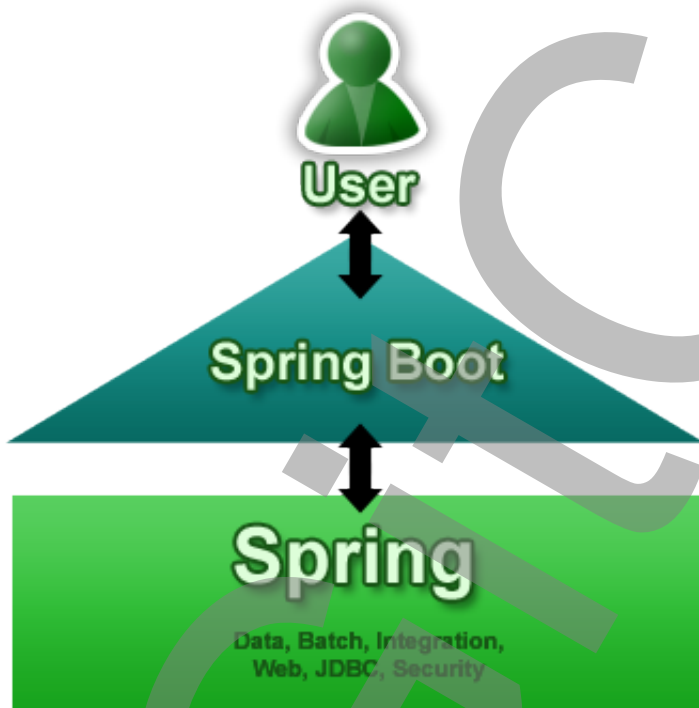
通過此課程的學習，相信你會完全掌握 Spring Boot 2.0 的使用，下面我們先來了解一下什麼是 Spring Boot？

## Spring Boot 介紹

Spring Boot 是由 Pivotal 團隊提供的全新框架，其設計目的是用來簡化新 Spring 應用的初始搭建以及開發過程，該框架使用了特定的方式來進行配置，從而使開發人員不再需要定義樣板化的配置。Spring Boot 默認配置了很多框架的使用方式，就像 Maven 整合了所有的 Jar 包，Spring Boot 整合了所有的框架。Spring Boot 的核心設計思想是：約定優於配置，Spring Boot 所有開發細節都是依據此思想進行實現。

Spring Boot 是一套全新的框架，它來自於 Spring 大家族，因此 Spring 所有具備的功能它都有並且更容易使用；Spring Boot 簡化了基於 Spring 的應用開發，通過少量的代碼就能創建一個獨立的、產品級別的 Spring 應用。

下图展示出 Spring Boot 在 Spring 生態中的位置：



該項目主要的目的是：

- 讓 Spring 的開發更快、更廣泛的上手；
- 使用默認方式實現快速開發；
- 提供大多數項目所需的非功能特性，諸如：嵌入式服務器、安全、心跳檢查、外部配置等。

## Spring Boot 特性

- 使用 Spring 项目引导页面可以在几秒构建一个项目；
- 方便对外输出各种形式的服务，如 REST API、WebSocket、Web、Streaming、Tasks；
- 非常简洁的安全策略集成；
- 支持关系数据库和非关系数据库；
- 支持运行期内嵌容器，如 Tomcat、Jetty；
- 强大的开发包，支持热启动；
- 自动管理依赖；
- 自带应用监控；
- 支持各种 IDE，如 IntelliJ IDEA、NetBeans。

## Spring Boot 市场热度

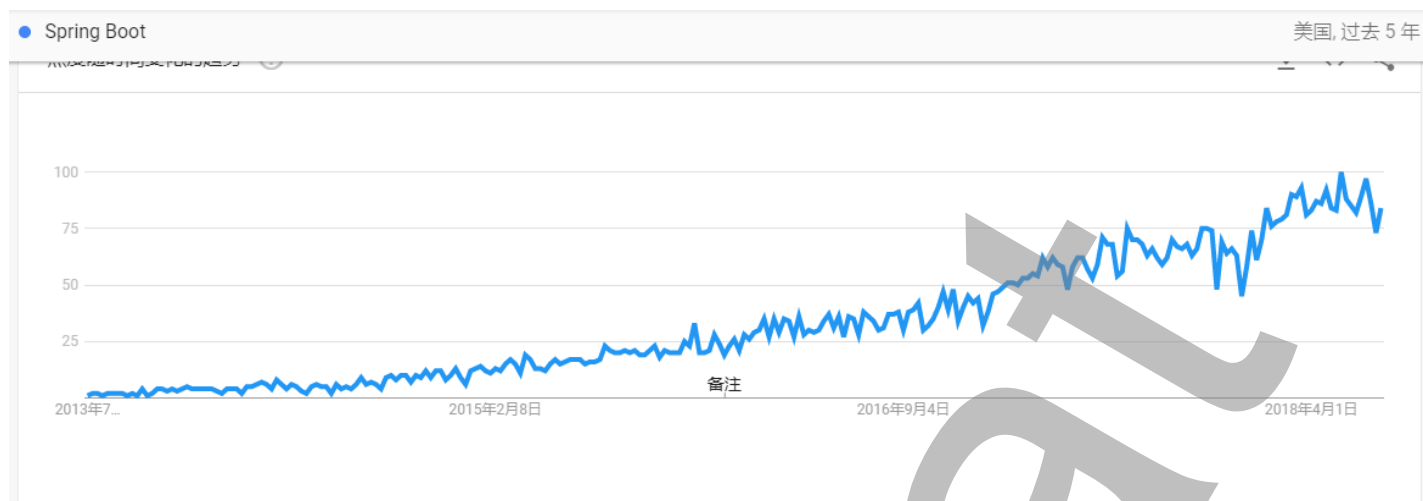
Spring Boot 于 2014 年发布了第一个正式版本，发布之后陆续有一些开源爱好者进行了研究，并迅速喜欢上了这款开源软件，Spring Boot 在初期低调快速的发展着。直到 2016 年 Spring Boot 才被真正使用了起来，期间很多研究 Spring Boot 的开发者，在网上写了大量关于 Spring Boot 的文章，推动了 Spring Boot 在行业内的的发展。

从 2016 年到 2018 年，是 Spring Boot 在中国发展的黄金时期，使用 Spring Boot 的企业和个人开发者越来越多，我们从 Spring Boot 关键字的百度指数可以看出。



上图为 2014 年到 2018 年 Spring Boot 的百度指数，可以看出 Spring Boot 2.0 的推出引发了搜索高峰。

通过谷歌趋势来看 Spring Boot 在美国的使用情况发现，中国和美国人民使用 Spring Boot 的整体频率保持一致，看来国内技术人同步全球的技术频率越来越快。



看到社区使用 Spring Boot 的热情，Spring 官方也非常重视 Spring Boot 的后续发展，已经将 Spring Boot 作为公司最顶级的项目来推广，放到了官网上第一的位置，后续 Spring Boot 的发展也被看好。



## Spring Boot 和微服务架构

微服务架构是在互联网高速发展，技术日新月异的变化以及传统架构无法适应快速变化等多重因素的推动下诞生的产物。互联网时代的产品通常有两类特点：需求变化快和用户群体庞大。在这种情况下，如何从系统架构的角度出发，构建灵活、易扩展的系统，快速应对需求的变化；同时，随着用户的增加，如何保证系统的可伸缩性、高可用性，成为系统架构面临的挑战。

如果还按照以前传统开发模式，开发一个大型而全的系统已经很难满足市场对技术的需求，这时候分而治之的思想被提了出来，于是我们从单独架构发展到分布式架构，又从分布式架构发展到 SOA 架构，服务不断地被拆分和分解，粒度也越来越小，直到微服务架构的诞生。

大约 2009 年开始，Netflix 完全重新定义了它的应用程序开发 and 操作模型，拉开了微服务探索的第一步，直到 2014 年 3 月 Martin Fowler 写的一篇文章 MicroServices 以更加通俗易懂的形式为大家定义了什么是微服务架构。Martin Fowler 在文中阐述了对微服务架构的设想，认为微服务架构是一种架构模式，它提倡将单一应用程序划分成一组小的服务，服务之间互相协调、互相配合，为用户提供最终价值。

Spring Boot 诞生时，正处于微服务概念在慢慢酝酿中，Spring Boot 的研发融合了微服务架构的理念，实现了在 Java 领域内微服务架构落地的技术支撑。Spring Boot 在开发、测试、部署、运维等方面都做了大量的优化，使用 Spring Boot 开发项目，可以快速响应需求、独立完成开发部署上线。

Spring Boot 的一系列特性容易实现微服务架构的落地，从目前众多的技术栈对比来看 Spring Boot 是 Java 领域微服务架构最优落地技术没有之一。

## 为什么学习 Spring Boot

Spring Boot 本身并不提供 Spring 框架的核心特性以及扩展功能，只是用于快速、敏捷地开发新一代基于 Spring 框架的应用程序。同时它集成了大量常用的第三方库配置（如 Redis、MongoDB、JPA、RabbitMQ、Quartz 等），Spring Boot 应用中的这些第三方库几乎可以零配置的开箱即用，大部分的 Spring Boot 应用都只需要非常少量的配置代码，开发者能够更加专注于业务逻辑。

使用 Spring Boot 开发项目，有以下几方面优势：

- Spring Boot 使开发变简单，Spring Boot 提供了丰富的解决方案，快速集成各种解决方案提升开发效率。
- Spring Boot 使配置变简单，Spring Boot 提供了丰富的 Starters，集成主流开源产品往往只需要简单的配置即可。
- Spring Boot 使部署变简单，Spring Boot 本身内嵌启动容器，仅仅需要一个命令即可启动项目，结合 Jenkins、Docker 自动化运维非常容易实现。
- Spring Boot 使监控变简单，Spring Boot 自带监控组件，使用 Actuator 轻松监控服务各项状态。

从软件发展的角度来讲，越简单的开发模式越会流行，简单的开发模式解放出更多生产力，让开发人员可以将精力集中在业务上，而不是各种配置、语法所设置的门槛上。Spring Boot 就是尽可能的简化应用开发的门槛。

Spring Boot 所集成的技术栈，几乎都是各互联网公司在使用的技术，跟着 Spring Boot 的路线去学习，基本可以了解国内外互联网公司的技术特点。

## 总结

Spring Boot 是一套快速开发框架，在微服务架构的大环境下 Spring Boot 一经推出就受到开源社区的关注。Spring Boot 有强大的生态整合能力，提供了众多的 Starters 包，非常方便 Spring Boot 集成第三方开源软件达到开箱即用的效果。Spring Boot 和微服务架构都是未来软件开发的一个大趋势，越早参与其中受益越大。