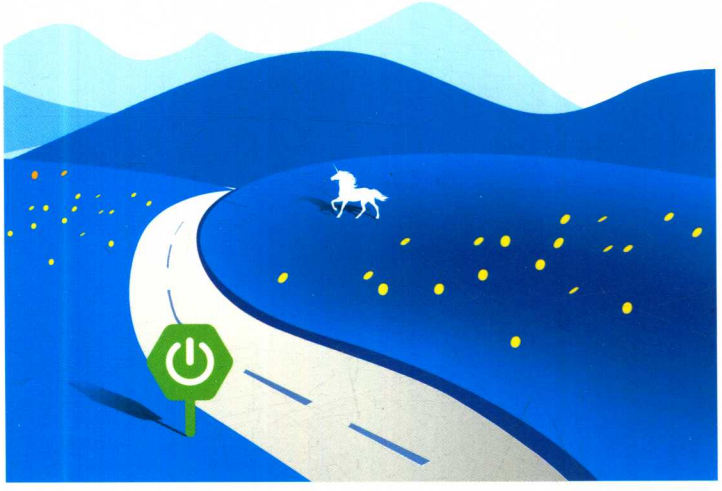
源代码下载



从0到1，决战Spring Boot

大老杨这本书，是我花了3天时间审校的。全书没有废话，一切从代码案例出发，记录了各种坑的解决方法，是Spring Boot初学者及核心技术巩固的最佳实践。

——泥瓦匠

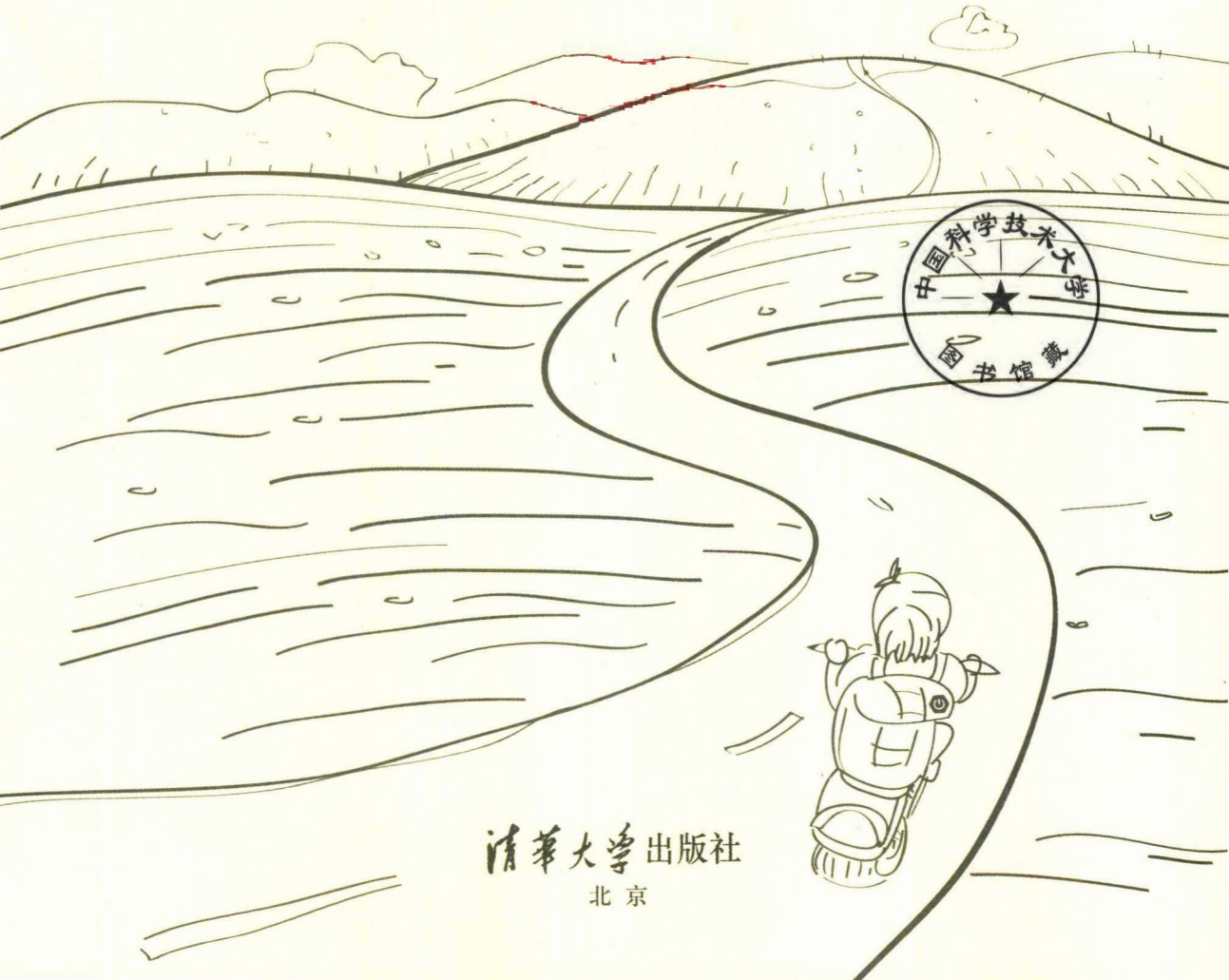
Spring Boot 2 实战之旅

杨 洋 著
泥瓦匠 审校

清华大学出版社

Spring Boot 2 实战之旅

杨洋 著



清華大學出版社
北京

内 容 简 介

Spring Boot 框架是目前微服务框架的最佳选择之一。本书采用 Spring Boot 2.0.3 版本讲解，从零起步系统深入地剖析了 Spring Boot 的核心技术。从功能点出发，每一章都是不同的 Spring Boot 应用之旅。全书分为 14 章，第 1 章和第 2 章是学习 Spring Boot 的入门阶段，从 Spring Boot 简介到开发环境部署等，让读者对 Spring Boot 有一个初步的认识；第 3 章到第 10 章是 Spring Boot 的融合阶段，介绍了 Spring Boot 搭建 Web 项目、操作数据库、使用缓存、日志、整合安全框架、结合消息队列和搜索框架，这些都是日常开发中一定会用到的内容，经过这个阶段的学习，会让读者熟练地运用 Spring Boot 进行敏捷开发。第 11 章和第 12 章是 Spring Boot 的拓展阶段，主要介绍了 Spring Boot 的一些常用的功能和如何在实际应用中的部署。第 13 章和第 14 章是 Spring Boot 的实战阶段，经过这两章的学习，使读者对 Spring Boot 的运用更加熟练，掌握实际项目的开发技能。

本书的特点是示例代码丰富，实用性和系统性较强，读者可以直接还原书中的示例。本书适用于初学者、Java 开发人员、Spring 爱好者和架构师。

图书在版编目（CIP）数据

Spring Boot 2 实战之旅/杨洋著. —北京：清华大学出版社，2019
ISBN 978-7-302-53162-3

Ⅰ. ①S… Ⅱ. ①杨… Ⅲ. ①JAVA 语言－程序设计Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2019）第 116039 号

责任编辑：王金柱
封面设计：王　翔
责任校对：闫秀华
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张**：24.25　　**字　　数**：621 千字
版　　次：2019 年 8 月第 1 版　　**印　　次**：2019 年 8 月第 1 次印刷
定　　价：79.00 元

产品编号：081783-01

前　言

微服务一词相信对很多开发者来说已经耳熟能详了。在我曾经工作的公司，还是使用单体项目来部署时，无论是打包还是运行都耗时耗力，这一直让我很苦恼。同时，每次需要创建新应用、构建项目配置 Spring 的时候也十分麻烦。一次偶然的情况，我接触了 Spring Boot 框架，开始对其“约定优先配置”的特性着迷了。这个由 Pivotal 团队进行维护开发的 Spring Boot，版本更迭非常快，社区活跃度很高。我在闲暇之余查阅了国内很多招聘网站，原来已经有很多公司将 Spring Boot 作为必备技能。

此后，我花费了很长的时间翻看技术博客、官方文档等，深入学习 Spring Boot 框架。在公司接下来的项目中，都以 Spring Boot 为主来构建项目，并且成功地将很多使用 Spring Boot 的项目投入生产，Spring Boot 框架的快速构建与部署与公司快速迭代版本的风格完美呼应。这是 Spring Boot 值得学习的一大原因。

本书沿袭我学习 Spring Boot 的路线，使用 Spring Boot 与当今常用的中间件结合，并且配备对应的实例代码。最后的两章项目实战是对 Spring Boot 的学习之路做出总结，为本书画上一个圆满的句号。希望读者阅读本书后能够有所收获。

如何阅读本书

在阅读本书的过程中，建议对照源代码按顺序学习。当然，如果对部分章节的内容比较熟悉，也可以直接跳过，学习需要巩固的章节。本书内容共分为 14 章，开发工具使用 IntelliJ IDEA，Spring Boot 版本为 2.0.3，各章节内容说明如下：

第 1 章介绍 Spring Boot 框架的特点以及学习它的重要性，最后列出 Spring Boot 的历史版本，让读者对 Spring Boot 有一个大致的了解。

第 2 章介绍如何搭建 Spring Boot 的开发环境，通过使用 IntelliJ IDEA 构建 Spring Boot 项目，并且对 Spring Boot 项目的基础结构进行介绍。

第 3 章介绍如何使用 Spring Boot 开发 Web 应用，了解 Spring MVC 和 Spring Web Flux 的不同，最后学习 Spring Boot 的一些 Web 模板框架，让读者可以对 Spring Boot 开发 Web 应用游刃有余。

第 4 章和第 5 章都是基于 Spring Boot 对数据库的使用进行学习。其中，第 4 章从 Spring Boot 使用各种数据库的依赖和配置开始介绍，然后介绍当今 Java 语言流行的 ORM 框架的使用，最后学习 Spring Boot 使用 Druid 数据库连接池。第 5 章介绍 Spring Boot 常用缓存框架，最后对 Redis 和 Memcached 进行比较，让读者选择缓存时有一定的基础。

第 6 章介绍 Spring Boot 对几种常用日志框架的使用，最后介绍分布式情况下如何使用 ELK 进行日志收集。

第 7 章介绍当今比较常用的两种安全框架，并且使用详细的案例对二者进行运用。

第 8 章介绍 Spring Boot 如何进行监控，涉及当今 Spring Boot 框架常用的监控，使读者对 Spring Boot 的运行状态更加了解。

第 9 章介绍 Spring Boot 如何使用消息队列，分别从 RabbitMQ、Kafka 和 RocketMQ 的使用实例进行介绍，最后对三者进行比较，让读者在选择消息队列时有一定的借鉴。

第 10 章对 Spring Boot 的两大常用搜索框架进行详细的介绍，从普通增、删、改、查到复杂查询，让读者使用搜索框架时不再茫然。

第 11 章介绍使用 Spring Boot 时的一些小技巧，比如启动 Banner、Lombok、邮件发送、事务、异常等。虽然知识略微零散，但是都是实用的技巧。

第 12 章介绍 Spring Boot 的多种部署方式，让读者可以根据实际情况部署自己的应用程序。

第 13 章和第 14 章分别使用博客系统和博客后台系统对 Spring Boot 的使用进行综合实战，这两个实战案例是对本书内容的总结。

本书读者对象

- 初学者
- Java 开发人员
- 架构师
- Spring 爱好者

本书技术支持

非常感谢大家能够购买和阅读本书。虽然完成本书尽了笔者最大的努力，但是由于笔者的精力和能力有限，在编写过程中难免会有一些疏漏和不足之外，希望各位读者不吝指正。关于本书的任何问题都可以发送电子邮件至 yangyang@dalaoyang.cn 与我交流。

源代码下载

本书所有源代码均上传至码云，地址是 https://gitee.com/dalaoyang/springboot_book。如果下载有问题，请发送电子邮件至 booksaga@126.com，邮件主题为“求 Spring Boot 2 实战之旅下载资源”。

致谢

在编写本书时，我得到了很多人的帮助。

首先，感谢我的妻子，在我遇到困难时给予鼓励，在我迷茫时的开导，谢谢她在我编写本书的过程中承担了所有家务，并且不遗余力地支持我。

其次，感谢我的父母，感谢他们从小对我的抚育与培养，感谢他们对我事业的支持。

另外，还需要感谢一下泥瓦匠在百忙之中对本书的细心校对，让本书的一些细节更加完善。

最后，感谢清华大学出版社的王金柱编辑，感谢您在本书编写、出版整个过程中的辛勤付出。也要感谢清华大学出版社所有参与本书编辑和出版的老师们，感谢大家对本书的帮助。

杨　洋

2019 年 3 月 1 日

目　录

第1章

Spring Boot 概述

本章将对 Spring Boot 进行整体的介绍，从 Spring Boot 的特点开始，逐步让大家了解学习 Spring Boot 会为我们带来的好处，最后从 Spring Boot 的发展史（Spring Boot 1.x 到 Spring Boot 2.x）来全面概述 Spring Boot 微服务框架的多功能性。

1.1 Spring Boot 简介

Spring Boot 是由 Pivotal 团队在 2014 年发布的全新框架。从 Spring Boot 的 Logo 中可以看到，Spring Boot 是要打造一个快速构建的 Spring 应用，如图 1-1 所示。正如 Spring 官网（官网地址：http://spring.io/）介绍的：

```
"Spring Boot is designed to get you up and running as quickly as possible,
with minimal upfront configuration of Spring. Spring Boot takes an opinionated
view of building production ready applications."
```

图 1-1 Spring Boot 官方 Logo 图片

翻译过来的意思大概是：Spring Boot 的设计是可以尽可能快地启动和运行，只需要最少的 Spring 配置。Spring Boot 对构建生产就绪应用程序具有独特的方式。从官方的介绍可以看出 Spring Boot 的核心思想是“约定优先于配置（Convention Over Configuration）”，其本质其实还是基于

Spring 来实现的。对于了解 Spring 的人或者使用过 Spring 的人来说，Spring 烦琐的配置让很多程序员眼花缭乱（各种 XML、Annotation 配置等），甚至很多时候发生错误也很难快速定位错误的地方。而在 Spring Boot 框架中，为我们提供了默认的配置，从而使开发人员不再需要定义样板化的配置，通过这种方式，Spring Boot 致力于在蓬勃发展的快速应用开发领域（Rapid Application Development）成为领导者。

1.2 Spring Boot 的特点

Spring 团队曾经为开发者提供了无数的便利，其提供的 IOC 和 AOP 两大特性一直为广大开发者所“深爱”。当然，Spring 框架还提供了很多优秀的特性，在这里就不一一介绍了。但是，在传统 Spring 框架中有一个重大的缺点，那就是在配置的时候很复杂，需要重复地进行一些配置。Spring 团队可能感受到了这一点，在 2014 年，Spring 团队发布了 Spring Boot 框架。另外，官网首页的 Spring Boot 部分也介绍了诸多 Spring Boot 的特点，如图 1-2 所示。本节将逐一介绍 Spring Boot 框架的特点。

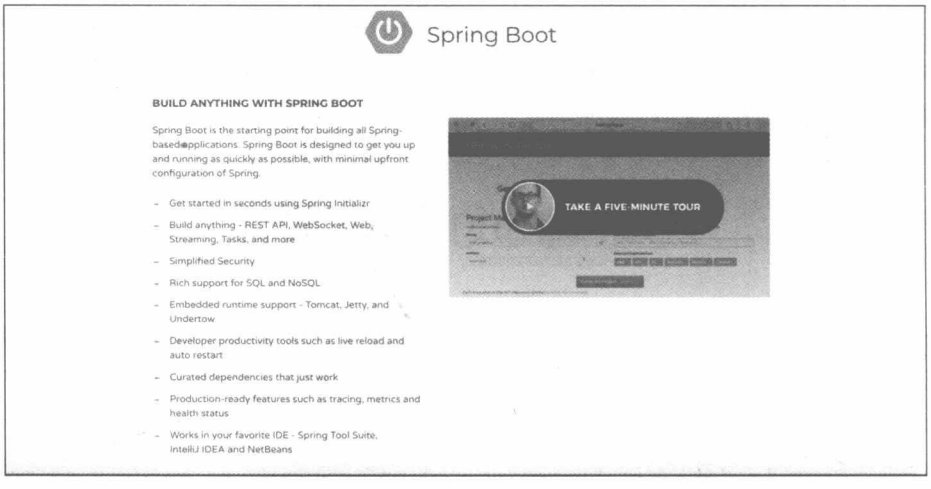


图 1-2 Spring Boot 官网简介（图片来源于 Spring 官网：http://spring.io/）

1.2.1 快速构建项目

Spring Boot 具有多种快速构建项目的方式，如下面几种形式：

（1）使用 Eclipse（MyEclipse）可以利用创建 Maven 项目的方式创建 Spring Boot 项目。当然，如果在 Eclipse 中安装了 Spring Tools，就可以直接创建 Spring Starter Project。

（2）使用 IntelliJ IDEA，可以利用创建 Spring Initializr 的方式创建 Spring Boot 项目，在后续章节会详细介绍这种方式的过程。

（3）使用 Spring Tool Suite，可以直接新建 Spring Starter Project 项目，过程类似 Eclipse 创建 Spring Boot 项目。

（4）使用官方文档创建项目，在Spring官方文档上面提供了一种在线生成Spring Boot项目的方式，首先访问Spring官方快速构建地址（官网地址：https://start.spring.io/），在这个页面上选择对应版本、构建工具等，填写完成后单击Generate Project按钮，即可在本地下载一个Spring Boot项目的压缩包。

当然，可能还有很多方式快速构建项目，这里就不一一介绍了。笔者在这里推荐使用IntelliJ IDEA开发项目，个人感觉这个开发工具还是很强大的，并且提供了很多插件供开发者使用，读者也可以根据自己的喜好进行选择，毕竟适合自己的才是最好的。

1.2.2 嵌入式Web容器

在传统Java Web项目中，当项目开发完成之后，还需要配置所需的Web容器（诸如Tomcat或者WebLogic之类的Web容器）。但是在Spring Boot搭建的项目中，内部提供了几种Web容器，如Tomcat、Jetty和Undertow。在Spring Boot 1.x中默认为Tomcat；Spring Boot 2.x中则分为两种情况，引入spring-boot-starter-web依赖为Tomcat，引入spring-boot-starter-webflux依赖则为Netty。当然，也支持使用之前指出的几种Web容器，开发者只需要根据场景选择适合的Starter来获取一个默认配置好的容器即可，当启动成功后，应用一个默认端口为8080的HTTP服务。

1.2.3 易于构建任何应用

Spring Boot提供了一个强大的starter依赖机制，实质上Spring团队将Spring Boot框架整合了一切常用的maven依赖，使Spring Boot想要整合对应依赖，就要将需要的依赖全部引入。比如，需要在项目中使用Web，也就是我们常说的Spring MVC，如果是原有的maven项目，就需要引入很多依赖才能完成这个简单的需求。但是在Spring Boot项目中，我们只需要在maven依赖中加入spring-boot-starter-web依赖即可，是不是很简单？这里再举一个例子，比如项目中需要使用MySQL数据库，这里只需要加入MySQL依赖，并且在配置文件中配置数据库信息就可以正常使用。

1.2.4 自动化配置

这个特点是上一个特点的延伸，在应用程序中引入依赖之后，其实还有一个强大之处在于Spring Boot应用会根据引入的依赖提供一些默认的配置供我们使用，如果需要修改，那么只需要在配置文件中修改对应的配置即可完成需求。这里还是以Spring MVC为例，传统Spring MVC项目需要配置对应的诸如ApplicationContext.xml（Spring配置文件）、ApplicationContext-mvc.xml（Spring MVC配置文件），而在Spring Boot中，这些需要的配置已经为我们默认配置了一套，不需要再进行配置了。比如，我们要加入Web应用程序根路径test的话，只需要在application.properties（Spring Boot应用程序默认配置文件）中加入server.servlet.context-path=/test即可。

1.2.5 开发者工具

在开发 Web 应用的时候，总会有一个困扰我们的问题，修改代码总是伴随不断重启项目，需要不断地断开 Web 容器，再重启来测试我们的代码。在 Spring Boot 应用中提供了开发者工具（spring-boot-devtools），当我们重新编译类文件的时候，开发者工具会自动替我们重启应用，无须手动单击重启。

1.2.6 强大的应用监控

在生产环境中，应用的各项指标监控更是必不可少。在 Spring Boot 应用中提供了一个 spring-boot-starter-actuator（以下简称 Spring Boot-Actuator）来供我们查看应用的各项指标，如 health（健康检查）、dump（活动线程）、env（环境属性）、metrics（内存，CPU 等）等指标，以监控我们的应用，同时可以配合使用 spring-boot-admin-starter-server（以下简称 Spring Boot-Admin）监控我们的项目。Spring Boot-Admin 可以在利用监控 Spring Boot-Actuator 端点的同时监控所有微服务应用的健康状态，如果出现异常，就可以向维护人员发送邮件或者以其他方式给予告警。不只是这样，就连监控神器 Prometheus 也可以通过简单的配置接入 Spring Boot 应用程序中。

1.2.7 默认提供测试框架

Spring Boot 应用在创建项目之后会默认为我们创建测试类的文件，实质上就是引入 spring-boot-starter-test 依赖，然后可以通过它对各种场景进行测试，足够满足对项目的测试需求。

1.2.8 可执行 Jar 部署

由于 Spring Boot 项目内嵌 Web 容器，因此提供了一种特殊部署方式，可以直接利用 Maven 或者 Gradle 对 Spring Boot 项目进行打包，生成一个 JAR 文件，然后直接在具备环境的服务器或本地环境中利用 java -jar xx.jar 执行 JAR 文件，使应用能够快速运行。

1.2.9 IDE 多样性

正如 1.2.1 小节介绍的，Spring Boot 支持的开发工具很多，无论是曾经几乎所有开发者都使用的 Eclipse 一族，还是现在流行的 IntelliJ IDEA，又或者是专门为开发 Spring 系列而生的 Spring Tool Suite 都是开发 Spring Boot 应用的不二法宝。

1.3　为什么要学习 Spring Boot

为什么要学习 Spring Boot 呢？这可能是很多读者心中的疑惑。在 1.2 节，我们通过了解 Spring Boot 的特点应该已经对 Spring Boot 框架产生了一定的兴趣，接下来笔者将从几个方面来整体阐述学习 Spring Boot 框架的理由。

1.3.1　简化工作

Spring Boot 最大的优点是在一定程度上简化了我们的工作，大致分为以下几个角度对工作进行简化。

- 依赖简化：Spring Boot 自有的 starter 中提供了一些可以快捷使用的依赖，让整合或集成一些常用的功能更便捷。
- 配置简化：在配置方法中，如果没有特殊情况，Spring Boot 为我们提供了一些默认的配置，比如端口号默认为 8080 等。
- 部署简化：正如前面介绍的可执行 JAR 部署，与传统服务的 Web 模式部署（打 WAR 包部署）相比，连安装 Web 容器的时间都节省了，不只是开发者，对运维人员也是福音。
- 监控简化：可以通过引用 Spring Boot 依赖的方式快捷提供监控端点，无代码侵入，十分便捷。

1.3.2　微服务时代

“微服务”一词最早是由 Martin Fowler 的《Microservices》一文（原文链接为 https://martinfowler.com/articles/microservices.html）提出的，其核心思想是将一个单体应用根据业务功能拆分成为多个服务，使业务代码之间不再耦合。接下来，我们介绍一下由单体应用转变为微服务应用的好处。

1. 微服务的优势

- 服务解耦：将单体应用转变为多个服务，服务与服务之间通过 HTTP 协议或其他约定好的协议等进行网络通信。
- 技术选型广泛：微服务不需要局限于固定的技术栈，各个服务的开发团队可以根据场景自由选择开发技术，如 Java、PHP、Node.Js 等。
- 服务并行开发：可以多个团队分别开发不同的模块，每个团队负责一个或者几个服务，相互不受影响。
- 单一职责：不同服务的团队只需要关注自己团队的业务，无须经常为了熟悉业务而耽误时间。
- 独立部署：由于每个服务都是独立的项目，因此当服务开发完成后，可以直接独立部署。
- 敏捷开发：每个服务的业务迭代只需要更新对应服务的功能即可，支持快速迭代。

- 故障隔离：在传统单体项目中，如果某个功能发生故障，就可能导致整个服务发生宕机，但是在微服务中，一个服务发生宕机，其他服务仍然可以继续工作。

2. 微服务的劣势

- 部署需要花费更多的精力：当服务拆分得非常多的时候，可能需要消耗更多的精力去运维管理这些应用。传统的单体应用下，运维人员只需要保证一个服务正常运行即可，但是拆分微服务后，可能需要保持几十，甚至上百、上千的服务高效运行，这对运维人员来说是一个很大的挑战。
- 服务间的接口问题：正因为拆分了微服务，服务与服务间使用接口进行相互调用，当一个接口需要改变格式或者增减数据时，所有调用这个接口的服务都需要做出相应的调整才能正确地使用。
- 高可用：拆分微服务后，可能有很多服务需要调用同一个服务或者接口，这个服务的可用性就需要让我们更加注意。
- 分布式事务：微服务系统各个服务间可能使用不同的数据库，比如搜索服务使用 Elasticsearch、基础服务使用 MySQL、评论服务使用 MongoDB，对于不同数据库间数据的一致性将是我们面临的重大挑战。
- 网络复杂性：由于各个服务间使用接口调用，因此系统间需要考虑很多网络延迟等客观因素来保证服务间的正常运转。
- 测试的复杂性：在测试方面，服务间的接口调用、服务间的测试需要一套整体的测试方案，自动化测试就显得必不可少。

由于 Spring Boot 项目可以提供快速开发、测试、部署，因此 Spring Boot 是微服务应用的不二选择。

1.3.3 社区背景强大

社区背景强大其实是 Spring Boot 如今盛行的原因。众所周知，Spring 家族对于开发者提供了无尽的便利，而作为 Spring 的亲儿子“Spring Boot”则继承了一切 Spring 的优点，并且规避了很多 Spring 框架臃肿的缺点，而后续 Spring 家族的分布式框架 Spring Cloud 也是基于 Spring Boot 框架实现的框架，所以作为 Spring 的爱好者，或者将要学习 Spring Cloud 框架的开发者，Spring Boot 是必须要学习的。

1.3.4 市场需求

在写这本书之前，笔者游历于各大国内、国外技术论坛，无论是在国内还是在国外，Spring Boot 的呼声都特别高，而且框架的更新频率特别快。如图 1-3 所示是从 2014 年到 2018 年 Spring Boot 的百度搜索指数。

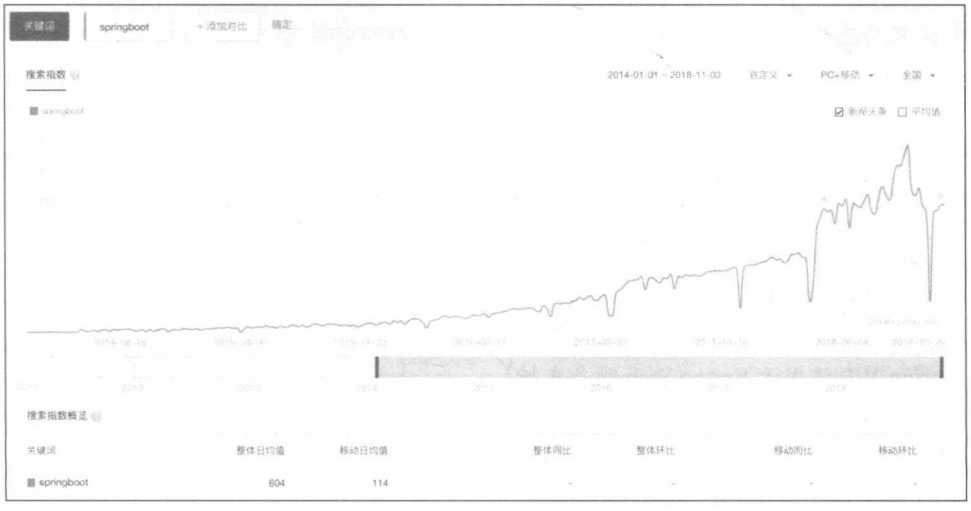


图 1-3　Spring Boot 百度搜索指数（图片来源于百度搜索指数）

从搜索指数可以看出，Spring Boot 的搜索值日趋增长，关注度特别高。

另外，我们从互联网招聘网站上来看，已经有超过 7 成以上的公司将 Spring Boot 框架作为筛选人员的必要条件，所以无论是从个人提升，还是比较实际的跳槽、涨薪等，学习 Spring Boot 都会为你的技术栈增光添彩。

1.4　Spring Boot 的发展历史

Pivotal 团队对于 Spring Boot 更新得非常频繁，而且在 Github 和国内社区的关注度都极高。接下来我们看一下 Spring Boot 的发展史。

1.4.1　发布里程碑（2013.8.6）

Phil Webb 在 Spring 官网博客上宣布了一个名为 Spring Boot 的新项目的第一个里程碑版本。

1.4.2　Spring Boot 1.0（2014.4）

Spring Boot 问世，为所有 Spring 开发提供快速和可广泛访问的入门体验，其中版本功能包括但不限于以下几点：

- 嵌入式服务器。
- 外部配置。
- 健康检查。
- 安全性。
- 快速运行。