微服务

微服务其实是服务化思路的一种最佳实践方向，遵循SOA（面向服务的架构）的思路，各个企业在服务化治理上面的道路已经走得很远了，整个软件交付链上各个环节的基础设置逐渐成熟了，微服务就诞生了。

微服务给我们也带来了很多挑战，服务“微”化之后，一个显著的特征就是服务的数量增多了。如果将软件开发和交付也作为一种生产模式的看待，那么数量众多的微服务实际上就类似于传统生产线上的产品，而在传统生产模式下，为了能够高效地生产大量产品，通常采用的就是标准化生产。

## 1.1简介

1. Spring boot是用来简化Spring应用程序的创建和开发过程；
2. Spring boot抛弃了繁琐的xml配置过程，采用大量的默认配置简化我们的开发过程；
3. 它让编码变简单了，配置变简单了，部署变简单了，监控变简单了；
4. 使用Spring Boot创建可以通过使用java -jar或更传统的战争部署启动的Java应用程序。我们还提供了一个运行“spring脚本”的命令行工具。

## 1.2Spring boot的特性

1. 快速创建基于Spring的应用程序；
2. 能直接使用java main方法启动内嵌的Tomcat，Jetty服务器运行Spring boot程序，不需要部署war包文件；
3. 提供约定的starter POM来简化Maven配置，让Maven的配置变得简单；
4. 根据项目的Maven依赖配置，Spring boot自动配置Spring、Spring mvc等；
5. 提供了程序的健康检查等功能；
6. 基本可以完全不使用XML配置文件，采用注解配置。

## 1.3Spring boot四大核心

1. **自动配置**：针对很多Spring应用程序和常见的应用功能，Spring Boot能自动提供相关配置；
2. **起步依赖**：告诉Spring Boot需要什么功能，它就引入需要的依赖库；
3. **Actuator**：让你能够深入运行中的Spring Boot应用程序，一探Spring Boot程序的内部信息；
4. **命令行界面**：这是Spring Boot的可选特性，主要针对Groovy语言的使用。

## 1.4系统要求

1. Spring Boot 2.1.8.RELEASE需要Java 8，并且与Java 12（包括）兼容。还需要Spring Framework 5.1.9.RELEASE或更高版本；
2. 构建工具

| Build Tool | Version |
| --- | --- |
| Maven | 3.3+ |
| Gradle | 4.4+ |

1. 支持以下嵌入式servlet容器：

| Name | Servlet Version |
| --- | --- |
| Tomcat 9.0 | 4.0 |
| Jetty 9.4 | 3.1 |
| Undertow 2.0 | 4.0 |

可以将Spring Boot应用程序部署到任何Servlet 3.1+兼容的容器中。