# 极速开发框架SpringBoot及源码分析

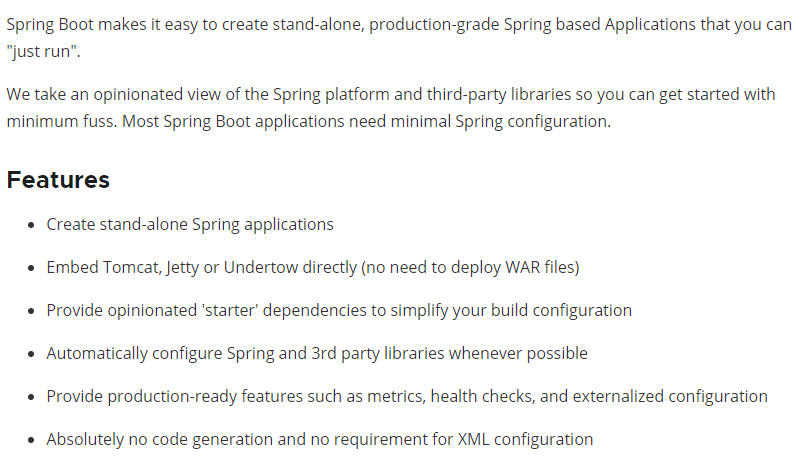
主讲：Cat 老师

**北京动力节点教育科技有限公司**  
**2020 • 北京**

**动力节点版权所有，禁止私自传播**

## What is SpringBoot？

源自官网：<https://spring.io/projects/spring-boot>



Spring Boot使开发独立的，产品级别的基于Spring的应用 变得非常简单，你只需"just run"；

我们为Spring平台及第三方库提供开箱即用的设置，这样你就可以有条不紊地开始，多数Spring Boot应用需要很少的Spring配置；

**特性：**

创建独立的Spring应用程序；

直接嵌入Tomcat，Jetty或Undertow（无需部署WAR文件）；

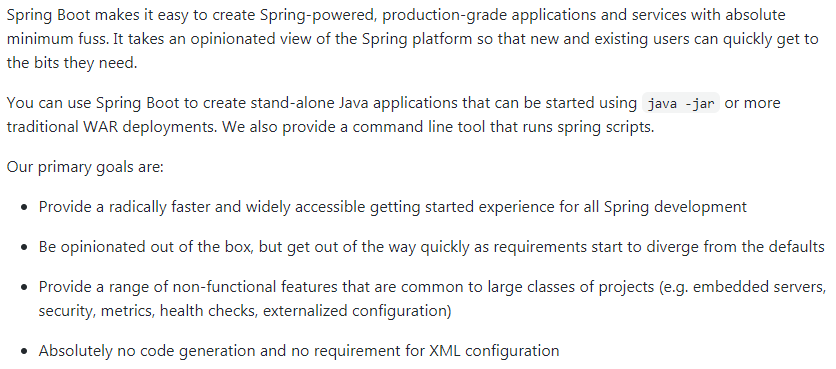
提供自以为是的“starter”依赖项，以简化构建配置；

尽可能自动配置Spring和3rd Party库；

提供可用于生产环境的功能，例如指标，运行状况检查和外部化配置；

完全没有代码生成，也不需要XML配置；

Github：<https://github.com/spring-projects/spring-boot>



通过Spring Boot，你可以轻松创建具有基于Spring的生产级应用程序和服务，而不必花太多功夫，SpringBoot是基于SpringFramework框架的，以便于新老用户都能快速上手。你可以使用Spring Boot创建独立的Java应用程序，然后通过java -jar运行，或者创建web项目，通过WAR部署启动，同时SpringBoot还提供了运行Spring脚本的命令行工具；

SpringBoot的主要目标是：

为所有Spring开发提供根本上更快速的入门开发体验（开发更快更方便）；

开箱即用，随着需求的变化，SpringBoot帮我们解决了很多常见的问题，比如快速集成其他组件；

提供一系列大型项目通用的非功能性功能（例如嵌入式服务器，安全性，性能指标，运行状况检查，外部化配置）

完全没有代码生成，也不需要XML配置；

总结一下就是：Spring家族推出springboot框架就是为了简化spring的配置与开发，使得在基于Spring框架的开发中开发起来更快速更方便，简化或减少相关的配置，提高开发效率；

## SpringBoot开发环境

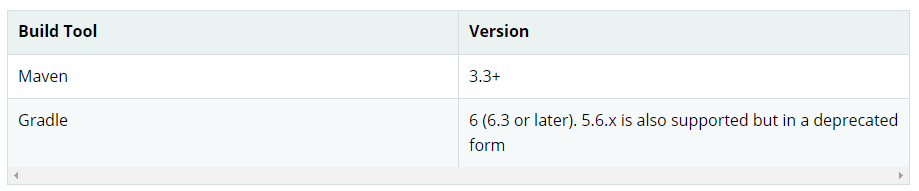
1、Spring Boot目前分为两大版本系列， 1.x系列和2.x系列，目前 Spring Boot 最新正式版为2.3.1.RELEASE；

2、如果是使用eclipse开发，推荐安装 Spring Tool Suite (STS) 插件；

3、如果使用 IDEA 旗舰版，自带了SpringBoot插件；

4、推荐使用 Java 8，Spring boot 1.x系列的版本兼容 Java 6，Spring boot 2.x系列需要至少Java 8；

5、Springboot2.X推荐使用 Maven 3.3+，太低的Maven版本会报错，Maven目前最新版本为 3.6.3；<http://maven.apache.org>



## SpringBoot Java Application

public static void main(String[] args) {

SpringApplication app = new SpringApplication(MySpringConfiguration.class);

app.setBannerMode(Banner.Mode.OFF);

app.run(args);

}

Fluent Builder API：

new SpringApplicationBuilder()  
 .sources(Application.class)  
 .bannerMode(Banner.Mode.*CONSOLE*)  
 .run(args);

*#关闭logo*spring.main.banner-mode=*off*spring.main.sources=com.bjpowernode.DemoApplication  
*#表示web项目*spring.main.web-application-type=*servlet*spring.main.allow-bean-definition-overriding=true  
server.port=8080

如何修改启动的logo日志：

在src/main/resources放入banner.txt文件

将新的logo文字粘贴到banner.txt文件中即可；

在 Spring Boot 框架中，要创建一个纯Java程序：

方式一：

1、SpringBoot开发纯Java程序，应该采用如下的起步依赖：

<!-- Springboot开发java项目的起步依赖 -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

2、直接在main方法中，根据SpringApplication.run()方法获取返回的Spring容器对象，再获取业务bean进行调用；

public static void main(String[] args) {

ConfigurableApplicationContext context = SpringApplication.run(SpringBootConsoleApplication.class, args);

HelloService helloService = (HelloService)context.getBean("helloService");

String hi = helloService.getMessage("hello, springboot");

System.out.println(hi);

}

方式二：

1、SpringBoot开发纯Java程序，应该采用如下的起步依赖：

<!-- Springboot开发java项目的起步依赖 -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

2、Spring boot 的入口类实现CommandLineRunner接口；

3、覆盖CommandLineRunner接口的run()方法，run方法中编写具体的处理逻辑即可；

@Autowired

private HelloService helloService;

@Override

public void run(String... args) throws Exception {

String ss = helloMessageService.getMessage("hello world!");

System.out.println(ss);

}

## 从SpringMVC到SpringBoot

我们通过一个案例，学习和掌握SpringBoot的开发；

## 创建SpringBoot项目

方式一： 采用IDEA 自带的插件创建 或者 使用eclipse的Spring Tool Suite (STS) 插件创建；(url路径有两个选择)

Spring官方：<https://start.spring.io/>

阿里云：<https://start.aliyun.com/>

方式二：直接使用 Maven 创建项目的方式创建，然后手动添加依赖；（无网络的情况下）

## Springboot项目结构及Maven依赖

也是一个标准的maven结构，多了一个前端模板技术的static和template文件夹；

## SpringBoot项目使用JSP

在Spring boot中使用jsp，按如下步骤进行：

1、在pom.xml文件中配置依赖项

*<!--前端页面使用JSP-->  
<!--引入Spring Boot内嵌的Tomcat对JSP的解析包-->*<**dependency**>  
 <**groupId**>org.apache.tomcat.embed</**groupId**>  
 <**artifactId**>tomcat-embed-jasper</**artifactId**>  
</**dependency**>  
*<!--jstl标签依赖的jar包start -->*<**dependency**>  
 <**groupId**>javax.servlet</**groupId**>  
 <**artifactId**>jstl</**artifactId**>  
</**dependency**>  
*<!--jstl标签依赖的jar包end -->*

2、在application.properties文件配置spring mvc的视图展示为jsp：

**spring.mvc.view.prefix**=**/  
spring.mvc.view.suffix**=**.jsp**

3、在src/main 下创建一个webapp目录，然后在该目录下新建jsp页面

4、Maven的pom.xml中配置页面编译位置

<**resource**>  
 <**directory**>src/main/webapp</**directory**>  
 <**targetPath**>META-INF/resources</**targetPath**>  
 <**includes**>  
 <**include**>\*\*/\*.\*</**include**>  
 </**includes**>  
</**resource**>

## SpringBoot集成MyBatis

Spring boot 集成 MyBatis的步骤如下：

1、在pom.xml中配置相关jar依赖；

*<!-- mybatis-spring-boot-starter -->*<**dependency**>  
 <**groupId**>org.mybatis.spring.boot</**groupId**>  
 <**artifactId**>mybatis-spring-boot-starter</**artifactId**>  
 <**version**>2.1.3</**version**>  
</**dependency**>

*<!-- MySQL的jdbc驱动包 -->*<**dependency**>  
 <**groupId**>mysql</**groupId**>  
 <**artifactId**>mysql-connector-java</**artifactId**>  
</**dependency**>

2、在Springboot的核心配置文件application.properties中配置数据源：

spring.datasource.username=root

spring.datasource.password=123456

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://192.168.172.128:3306/workdb?**serverTimezone=GMT%2B8&**useUnicode=true&characterEncoding=utf8&useSSL=false

3、通过mybatis反向工程生成mapper接口和model类；

4、在MyBatis的Mapper接口中添加@Mapper注解；

或者 在运行的主类上添加

@MapperScan("com.bjpowernode.mapper") 注解包扫描；

1. 接下来就是按照ssm框架的方式编写代码即可；

注意：Idea当你从其他地方粘贴一个文件过来的时候，它往往是不编译的，需要手动编译；

## SpringBoot增删改查

和原来的SSM项目开发一致；

## SpringBoot项目热部署插件

热部署是指当我们修改代码后，服务能自动重启加载新修改的内容，这样大大提高了我们开发的效率；

Spring boot热部署通过添加一个插件实现；

插件为：spring-boot-devtools，在Maven中配置如下：

<!-- springboot 开发自动热部署 -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<optional>true</optional>

</dependency>

该热部署插件在实际使用中会有一些小问题，特别是分布式开发，比如dubbo开发框架，有点问题，需要手动重启；

## Idea之JRebel热部署插件

Jrebel是一个收费插件，需要破解，该插件可以实现热部署，具体破解可以参考网上的一些方法：

1、首先生成一个GUID，可以用该网址生成：<https://www.guidgen.com/>

2、用该网址https://jrebel.qekang.com/ + 生成的 GUID 激活；

例如:

https://jrebel.qekang.com/0b2e2ca8-6dde-47a9-b9da-f74c7e5bcff2

3、在Idea中设置为离线模式，来防止失效；

File -> Settings -> JRebel -> [Work offline] 按钮点一下；