

# SpringBoot篇

## 0、什么是Springboot?

Spring Boot 可以轻松创建可以“直接运行”的独立的、生产级的基于 Spring 的应用程序。Spring Boot 应用程序 只需要最少的 Spring 配置。

特征

- 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 “starter” 依赖项以简化 项目的构建配置
- 尽可能自动配置 Spring 和 第三方库
- 提供生产就绪功能，例如指标、健康检查和外部化配置
- 完全无需代码生成，无需 XML 配置

入门

## 1、为什么要用SpringBoot?

优点非常多，如：

1. 独立运行：内嵌了各种servlet容器，Tomcat、Jetty等，不再需要打成war包部署到容器中，**只要打成一个可执行的jar包就能独立运行**，所有的依赖包都在一个jar包内。
2. 简化配置：spring-boot-starter-web 启动器自动依赖其他组件，简少了maven的配置。
3. 自动配置：能根据当前类路径下的类、jar包来自动配置bean，比如添加一个spring-boot-starter-web启动器就能拥有web的功能，无需其他配置。
4. 无代码生成和XML配置：配置过程中无代码生成，也无需XML配置文件就能完成所有配置工作，这一切都是借助于条件注解完成的，这也是Spring4.x的核心功能之一。
5. 应用监控：Springboot提供一系列端点可以监控服务及应用，做健康检测。

## 2、SpringBoot 的核心注解是哪个？它主要由哪几个注解组成的？

启动类上面的注解是 `@SpringBootApplication`，它也是 SpringBoot 的核心注解，主要组合包含了以下 3 个注解：

- `@SpringBootConfiguration`：组合了 `@Configuration` 注解，实现配置文件的功能。允许在上文中注册额外的bean或导入其他配置类
- `@EnableAutoConfiguration`：打开自动配置的功能，也可以关闭某个自动配置的选项，如关闭数据源自动配置功能：`@SpringBootApplication(exclude = { DataSourceAutoConfiguration.class})`。
- `@ComponentScan`：Spring组件扫描。

```
/**
 * @see org.springframework.boot.autoconfigure.SpringBootApplication
 */
```

### 3、SpringBoot运行有哪几种方式？

1. 打包用命令或者放到容器中运行
2. 用 Maven/Gradle 插件运行
3. 直接执行 main 方法运行

### 4、如何理解 SpringBoot 中的 Starters？

Starters可以理解为启动器，它包含了一系列可以集成到应用里面的依赖包，你可以一站式集成Spring及其他技术，而不需要到处找示例代码和依赖包。如你想使用Spring JPA访问数据库，只要 加入spring-boot-starter-data-jpa 启动器依赖就能使用了。Starters包含了许多项目中需要用到的依赖，它们能快速持续的运行，都是一系列得到支持的管理传递性依赖。

Starters 命名规则：官方的启动器都是以spring-boot-starter-命名的，代表了一个特定的应用类型。第三方的启动器不能以spring-boot开头命名，它们都被官方保留。一般一个第三方的应该这样命名，像mybatis的 mybatis-spring-boot-starter。

Starters分类：

1. 应用类启动器
2. 生产启动器
3. 技术类启动器
4. 其他第三方启动器

### 5、SpringBoot如何在启动的时候运行一些特定的代码？

如果你想在启动的时候运行一些特定的代码，你可以

实现接口 `ApplicationRunner` 或者 `CommandLineRunner`，这两个接口实现方式一样，它们都只提供了一个run方法。

CommandLineRunner：启动获取命令行参数

```
/**
 * @see org.springframework.boot.ApplicationRunner
 * @see org.springframework.boot.CommandLineRunner
 */
```

### 6、SpringBoot需要独立的容器运行吗？

可以不需要，内置了 Tomcat/ Jetty 等容器。

### 7、SpringBoot中的监视器是什么？

`spring boot actuator` 是spring启动框架中的重要功能之一。监视器可帮助 开发者访问生产环境中正在运行的应用程序的当前状态。有几个指标必须在生产环境中进行检查和监控。

即使一些外部应用程序可能正在使用这些服务来向相关人员触发警报消息。监视器模块公开了一组可直接作为HTTP URL访问的REST端点来检查状态。

## 8、SpringBoot如何使用实现异常处理？

Spring提供了一种使用 ControllerAdvice 处理异常的非常有用的方法。我们通过实现一个 ControllerAdvice 类，来处理控制器类抛出的所有异常。

```
/**
 * @see org.springframework.web.bind.annotation.ControllerAdvice
 */
```

## 9、你如何理解 SpringBoot 中的 Starters？

Starters可以理解为启动器，它包含了一系列可以集成到应用里面的依赖包，你可以一站式集成Spring及其他技术，而不需要到处找示例代码和依赖包。

如你想使用 Spring JPA 访问数据库，只要加入 spring-boot-starter-data-jpa 启动器依赖就能使用了。

## 10、springboot常用的starter有哪些

spring-boot-starter-web 嵌入tomcat和web开发需要servlet与jsp支持

spring-boot-starter-data-jpa 数据库支持

spring-boot-starter-data-redis redis数据库支持

mybatis-spring-boot-starter 第三方的mybatis集成starter

spring-boot-starter-data-solr solr支持

## 11、SpringBoot 实现热部署有哪几种方式？

主要有两种方式：

```
Spring Loaded
Spring-boot-devtools
```

## 12、如何理解 Springboot 配置加载顺序？

在 里面，可以使用以下几种方式来加载配置。

1. properties 文件
  2. YAML文件
  3. 系统环境变量
  4. 命令行参数
- 等等。...

## 13、SpringBoot 的核心配置文件有哪几个？它们的区别是什么？

SpringBoot 的核心配置文件是 application 和 bootstrap 配置文件。配置文件类型是 properties 或 yml

application 配置文件 主要用于 项目的自动化配置。

bootstrap 配置文件有以下几个应用场景。

使用 Spring Cloud Config 配置中心时，这时需要在 bootstrap 配置文件中添加连接到配置中心的配置属性来加载外部配置中心的配置信息；

- 一些固定的不能被覆盖的属性；
- 一些加密/解密的场景；