**1、Spring Boot和Spring MVC**

* Spring是一个“引擎”；
* Spring MVC是基于Spring的一个MVC框架 ；
* Spring Boot是基于Spring4的条件注册的一套快速开发整合包 。

1. **为什么选择spring boot**

传统Spring MVC开发一个简单的Hello World Web应用程序:

1）、一个项目结构，其中有一个包含必要依赖的Maven或者Gradle构建文件，最起码要有Spring MVC和Servlet API这些依赖。

2）、一个web.xml文件（或者一个WebApplicationInitializer实现），其中声明了Spring的DispatcherServlet。

3）、一个启动了Spring MVC的Spring配置

4）、一控制器类，以“hello World”相应HTTP请求。

5）、一个用于部署应用程序的Web应用服务器，比如Tomcat。

这份清单里面只有一个东西是和Hello World功能相关的，即控制器，剩下的都是Spring开发的Web应用程序必需的通用模板。

Spring 最初利用“工厂模式”（DI）和“代理模式”（AOP）解耦应用组件。大家觉得挺好用，于是按照这种模式搞了一个 MVC 框架（一些用Spring解耦的组件），用开发web应用（SpringMVC）。然后有发现每次开发都要搞很多依赖，写很多样板代码很麻烦，于是搞了一些懒人整合包（starter），这套就是Spring Boot。   
**3、Spring Boot的介绍**

从本质上来说，Spring Boot就是Spring,它做了那些没有它你也会去做的Spring Bean配置。它使用“**习惯优于配置**”（项目中存在大量的配置，此外还内置了一个习惯性的配置，让你无需手动进行配置）的理念让你的项目快速运行起来。使用Spring Boot很容易创建一个独立运行（运行jar,内嵌Servlet容器）、准生产级别的基于Spring框架的项目，使用Spring Boot你可以不用或者只需要很少的Spring配置。

最重要的是以下四个核心。

* **[自动配置（auto-configuration）](http://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-auto-configuration.html)**：针对很多Spring应用程序常见的应用功能，Spring Boot能自动提供相关配置。
* **起步依赖（starters）**：告诉Spring Boot需要什么功能，它就能引入需要的库。
* **命令行界面（cli）**：这是Spring Boot的可选特性，借此你只需写代码就能完成完整的应用程序，无需传统项目构建。
* **Actuator**：让你能够深入运行中的Spring Boot应用程序，actuator是spring boot提供的对应用系统的自省和监控的集成功能，可以对应用系统进行配置查看、相关功能统计等

Spring Boot使用“习惯优于配置”的理念让你的项目快速运行起来。所以，我们要想把Spring Boot玩的溜，就要懂得如何开启各个功能模块的默认配置，这就需要了解Spring Boot的配置文件application.properties，可以参照官方文档

[Spring Boot官网](https://spring.io/)

[快速搭建Spring Boot项目](http://projects.spring.io/spring-boot/)

[Spring Boot官方文档](http://docs.spring.io/spring-boot/docs/1.5.4.RELEASE/reference/html/)