

spring、springboot、springmvc的区别

笔记本: spring

创建时间: 2019/5/3 21:41

更新时间: 2019/5/3 23:12

作者: ANNER

URL: <https://www.jianshu.com/p/42620a0a2c33>

spring

spring是一个开源的应用程序框架，提供了一个简易的开发方式，通过这种开发方式，将避免那些可能致使代码变得繁杂混乱的大量的业务/工具对象

核心思想: IOC

什么是依赖注入

- 依赖注入(Dependency Injection)和控制反转(Inversion of Control)是同一个概念。具体含义是:当某个角色(可能是一个Java实例，调用者)需要另一个角色(另一个Java实例，被调用者)的协助时，在传统的程序设计过程中，通常由调用者来创建被调用者的实例。但在Spring里，创建被调用者的工作不再由调用者来完成，因此称为控制反转;创建被调用者实例的工作通常由Spring容器来完成，然后注入调用者，因此也称为依赖注入。

依赖注入的好处

- Spring 能有效地组织J2EE应用各层的对象。不管是控制层的Action对象，还是业务层的Service对象，还是持久层的DAO对象，都可在Spring的管理下有机地协调、运行。Spring将各层的对象以松耦合的方式组织在一起，Action对象无须关心Service对象的具体实现，Service对象无须关心持久层对象的具体实现，各层对象的调用完全面向接口。当系统需要重构时，代码的改写量将大大减少。

spring mvc

Spring MVC是Spring的一部分，Spring 出来以后，大家觉得很好用，于是按照这种模式设计了一个 MVC框架（一些用Spring 解耦的组件），主要用于开发WEB应用和网络接口，它是Spring的一个模块，通过Dispatcher Servlet, ModelAndView 和 View Resolver，让应用开发变得很容易

springboot

Spring社区推出了Spring Boot，它的目的在于实现自动配置，降低项目搭建的复杂度，如需要搭建一个接口服务，通过Spring Boot，几行代码即可实现

```
//引入spring-boot-starter-web依赖
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

```
//声明Spring Boot应用，直接写业务逻辑即可
@Controller
@SpringBootApplication
public class MockServerApplication {
    @RequestMapping("/hi")
    @ResponseBody
    String home() {
        return "how are you!";
    }

    public static void main(String[] args) {
        SpringApplication.run(MockServerApplication.class, args);
    }
}
```

你甚至都不用额外的WEB容器，直接生成jar包执行即可，因为spring-boot-starter-web模块中包含有一个内置tomcat，可以直接提供容器使用；基于Spring Boot，不是说原来的配置没有了，而是Spring Boot有一套默认配置，我们可以把它看做比较通用的约定，而Spring Boot遵循的也是**约定优于配置原则**，同时，如果你需要使用到Spring以往提供的各种复杂但功能强大的配置功能，Spring Boot一样支持

三者比较

Spring MVC和Spring Boot都属于Spring，Spring MVC 是基于Spring的一个 MVC 框架，而Spring Boot 是基于Spring的一套快速开发整合包



