

1、什么是 Spring Boot ?

Spring Boot 是 Spring 开源组织下的子项目，是 Spring 组件一站式解决方案，主要是简化了使用 Spring 的难度，简省了繁重的配置，提供了各种启动器，开发者能快速上手。

2、Spring Boot 有哪些优点？

减少开发，测试时间和努力。

使用 JavaConfig 有助于避免使用 XML。

避免大量的 Maven 导入和各种版本冲突。

提供意见发展方法。

通过提供默认值快速开始开发。

没有单独的 Web 服务器需要。这意味着你不再需要启动 Tomcat , Glassfish 或其他任何东西。

需要更少的配置 因为没有 web.xml 文件。只需添加用@

Configuration 注释的类 然后添加用@Bean 注释的方法 Spring

将自动加载对象并像以前一样对其进行管理。您甚至可以将

@Autowired 添加到 bean 方法中，以使 Spring 自动装入需要的依赖关系中。基于环境的配置 使用这些属性，您可以将您正在使

用的环境传递到应用程序：-Dspring.profiles.active =

{enviornment}。在加载主应用程序属性文件后，Spring 将在

(application{environment}.properties) 中加载后续的应用程序属性文件。

3、Spring Boot 的核心配置文件有哪几个？它们的区别是什么？

Spring Boot 的核心配置文件是 application 和 bootstrap 配置文件。

application 配置文件这个容易理解，主要用于 Spring Boot 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/解密场景

4、Spring Boot 的配置文件有哪几种格式？它们有什么区别？

.properties 和 .yaml，它们的区别主要是书写格式不同。

1, properties

```
app.user.name = javastack
```

2, yaml

```
app:  
  user:  
    name: javastack
```

5、Spring Boot 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 Spring

Boot 的核心注解 主要组合包含了以下 3 个注解：

1. @SpringBootConfiguration：组合了 @Configuration 注解，实现配置文件的功能。

2. @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 })。

3. @ComponentScan：Spring 组件扫描。

6、开启 Spring Boot 特性有哪几种方式？

1. 继承 spring-boot-starter-parent 项目

2. 导入 spring-boot-dependencies 项目依赖

7、Spring Boot 需要独立的容器运行吗？

可以不需要，内置了 Tomcat/ Jetty 等容器。

8、运行 Spring Boot 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

9、Spring Boot 自动配置原理是什么？

注解 `@EnableAutoConfiguration`, `@Configuration`,

`@ConditionalOnClass` 就是自动配置的核心,首先它得是一个配

置文件,其次根据类路径下是否有这个类去自动配置。

10、Spring Boot 2.X 有什么新特性?与 1.X 有什么区别?

配置变更

JDK 版本升级

第三方类库升级

响应式 Spring 编程支持

HTTP/2 支持

配置属性绑定

更多改进与加强...

11, 如何使用 Spring Boot 实现分页和排序?

使用 Spring Boot 实现分页非常简单。使用 Spring Data-JPA 可以实现将可分页的

`org.springframework.data.domain.Pageable` 传递给存储库方法。

12, 如何实现 Spring Boot 应用程序的安全性?

为了实现 Spring Boot 的安全性,我们使用

`spring-boot-starter-security` 依赖项,并且必须添加安全配置。

它只需要很少的代码。配置类将必须扩展

WebSecurityConfigurerAdapter 并覆盖其方法。

13 , 如何集成 Spring Boot 和 ActiveMQ ?

对于集成 Spring Boot 和 ActiveMQ , 我们使用

spring-boot-starter-activemq 依赖关系。 它只需要很少的配置, 并且不需要样板代码。

14、什么是 YAML ?

YAML 是一种人类可读的数据序列化语言。它通常用于配置文件。

与属性文件相比, 如果我们想要在配置文件中添加复杂的属性,

YAML 文件就更加结构化, 而且更少混淆。可以看出 YAML 具有

分层配置数据。

15 , Spring Boot 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。

Spring boot 监视器可帮助您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。

16 , 什么是 Swagger ? 你用 Spring Boot 实现了它吗 ?

Swagger 广泛用于可视化 API，使用 Swagger UI 为前端开发人员提供在线沙箱。Swagger 是用于生成 RESTful Web 服务的可视化表示的工具，规范和完整框架实现。它使文档能够以与服务器相同的速度更新。当通过 Swagger 正确定义时，消费者可以使用最少量的实现逻辑来理解远程服务并与其进行交互。因此，Swagger 消除了调用服务时的猜测。

17,如何使用 Spring Boot 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。

18,RequestMapping 和 GetMapping 的不同之处在哪里？

RequestMapping 具有类属性的，可以进行 GET,POST,PUT 或者其它的注释中具有的请求方法。

GetMapping 是 GET 请求方法中的一个特例。它只是 ResquestMapping 的一个延伸，目的是为了提高清晰度。

19、Spring Boot 可以兼容老 Spring 项目吗，如何做？

可以兼容，使用 @ImportResource 注解导入老 Spring 项目配置文件。

20、保护 Spring Boot 应用有哪些方法？

在生产中使用 HTTPS 使用 Snyk 检查你的依赖关系 升级到最新

版本 启用 CSRF 保护 使用内容安全策略防止 XSS 攻击