**SpringBoot框架技术知识点**

1. 什么是Spring Boot？
2. Spring Boot、Spring MVC 和 Spring 有什么区别？
3. 什么是 Spring Boot Stater ？
4. 谈谈你对Maven的理解。
5. 什么是YAML？
6. 创建一个 Spring Boot Project 的步骤是什么？
7. SpringBoot的运行原理？
8. @SpringBootApplication相当于那几个类的注解集成？
9. SpringBoot的常用注解。
10. springboot读取配置文件的方式？
11. springboot集成mybatis的过程？
12. 如何在 Spring Boot 中添加通用的 JS 代码？
13. 开启 Spring Boot 特性有哪几种方式？

**建议答案**

1. springboot是一种快速使用spring框架的简便方式，springboot简单来说相当于一个程序搭建的脚手架，最大的作用就是能够帮助我们快速构建项目，并且尽可能的减少xml的配置文件，让程序的开发更加的简单，开箱即用，快速上手，让我们在程序开发的过程中能够将更多的精力去关注到程序的业务逻辑，而并非是配置文件
2. 1、Spring

Spring最重要的特征是依赖注入。所有 SpringModules 不是依赖注入就是 IOC 控制反转。当我们恰当的使用 DI 或者是 IOC 的时候，我们可以开发松耦合应用。松耦合应用的单元测试可以很容易的进行。

2、Spring MVC

Spring MVC 提供了一种分离式的方法来开发 Web 应用。通过运用像DispatcherServelet，MoudlAndView 和 ViewResolver 等一些简单的概念，开发 Web应用将会变的非常简单。

3、SpringBoot

Spring 和 SpringMVC 的问题在于需要配置大量的参数。Spring Boot 通过一个自动配置和启动的项来目解决这个问题。为了更快的构建产品就绪应用程序，Spring Boot 提供了一些非功能性特征。

1. 启动器是一套方便的依赖没描述符，它可以放在自己的程序中。你可以一站式的获取你所需要的 Spring 和相关技术，而不需要依赖描述符的通过示例代码搜索和复制黏贴的负载。例如，如果你想使用 Sping 和 JPA 访问数据库，只需要你的项目包含 spring-boot-starter-data-jpa 依赖项，你就可以完美进行。
2. 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。
3. Maven就是一个包含了项目对象模型(Project Object Model, POM)的软件项目管理工具，可以通过配置描述信息来管理项目的构建，报告和文档。该工具可以帮助程序员从繁琐的项目配置工作中解放出来，轻松地进行工程构建、Jar包管理、代码编译，自动运行单元测试、打包、生成报表，甚至还能部署项目、生成Web站点
4. 1）可以通过maven来新建项目，利用pom文件导入项目所需的包。

2）通过https://start.spring.io/ 提供的模板，直接生成到下载到本地，初学者，推荐用该种方法，比较简单。

3）项目导入（若用第2种方法）到eclispe中，具体方法，参考教材，

4）导入后，如果发现pom文件的错误提示（第1行打”X”),一般情况下，不影响使用，也可不理它。当然也有解决办法，这种错误往往由于各种软件的JDK版本不一样引起的，需要统一设置，或加插件。

1. 1）自动初始化，包括自动扫描有注解的包，而无须配置，导入到系统的IOC容器中。

2）到自动执行具有@config的类，导入到系统的IOC容器中。

3）自动添加Tomcat和Spring MVC的依赖，并进行自动配置

4）自动启动服务器（TOMCAT）

1. 1）@Configuration：它是Spring的注解，实际就是JavaConfig形式的SpringIOC容器的配置类。

2）@ComponentScan：组件扫描（包的路径）的意思，在Spring中很重要，对应以xml配置文件中<context:component-scan>元素，用户代码放在启动类所在根目录下，则无须配置，自动扫描。

3）@EnableAutoConfiguration：该注解是借助@Import的帮助，将所有符合自动配置条件的bean定义加载到IOC容器中，并会对Tomcat和Spring MVC进行自动配置。

1. 
2. springboot默认读取配置文件为application.properties或者是application.yml
3. 添加mybatis的starter maven依赖

<dependency>

<groupId>org.mybatis.spring.boot</groupId>

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

<version>1.3.2</version>

</dependency>

在mybatis的接口中 添加@Mapper注解

在application.yml配置数据源信息

1. 在源文件夹下，创建一个名为 static 的文件夹。然后，你可以把你的静态的内容放在这里面。例如，myapp.js 的路径是 resources\static\js\myapp.js
2. 1）继承spring-boot-starter-parent项目

2）导入spring-boot-dependencies项目依赖

·················

············································································································································································································································································································································································································································································································································································································································································································································································································································································································································································································································································································································································································