######################SpringBoot常见面试题#####################

## 了解SpringBoot自动配置原理

* 各种配置拥有默认值

默认配置最终都是映射到XXXProperties

配置文件的值最终会绑定每个类上，这个类会在容器中创建对象

* SpringBoot所有的自动配置功能都在spring-boot-autoconfigure包里面
* 按需加载所有自动配置
  + - 非常多的start
    - 引入哪些场景这个场景的自动配置才会开启
* @SpringBootConfiguration
  + @Configuration.代表当前是一个配置类
* @ComponentScan 指定扫描哪些 spring注解
* @EnableAuoConfiguration

当spring boot扫描到@EnableAutoConfiguration注解时则会将

spring-boot-autoconfigure.jar/META-INF/spring.factories文件中

org.springframework.boot.autoconfigure.

对应的value里的所有xxxConfiguration类加载到IOC容器中。xxxAutoConfiguration

类一般都会有@ConditionalOnxxx注解，通过这些条件注解来判断是否真正的创建xxxConfiguration对象。

spring boot的各种spring-boot-starter-xxx.jar也正是居于此注解来达到自动装配的目的

## SpringBoot、Spring MVC和Spring有什么区别？

* Spring

Spring最重要的特征是依赖注入。所有Spring Modules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，可以开发松耦合应用。

* 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。

* SpringBoot

Spring和Spring MVC的问题在于需要配置大量的参数。

SpringBoot通过一个自动配置和启动的项来解决这个问题。

## 自动配置原理

总结：

·SpringBoot先加载所有自动配置类， XXXAutoConfiguration

·每个自动配置类按照条件进行生效，默认都会绑定配置文件指定的值，xxxProperties里面拿，xxxProperties和

和配置文件进行绑定

·生效的配置类就会给容器中装配很多组件

·只要容器中有这些组件，相当于功能就有了

·定制化配置

**用户直接直接@Bean替换底层的组件**

**用户去看这个组件是获取配置文件所对应的前缀去配置文件中修改**

Spring Boot 是微服务中最好的 Java 框架。为了更好的理解与使用SpringBoot，同时，为了记录平时的点点滴滴，便于日后的面试！

1、什么是SpringBoot？

用来简化Spring应用的初始搭建以及开发过程，使用特定的方式来进行配置

创建独立的Spring引用程序main方法运行

嵌入的tomcat无需部署war文件

简化maven配置

自动配置Spring添加对应的功能starter自动化配置

SpringBoot来简化Spring应用开发，约定大于配置，去繁化简

2、SpringBoot有哪些优缺点？

优点

独立运行

Spring Boot 而且内嵌了各种 servlet 容器，Tomcat、Jetty 等，现在不再需要打成war 包部署到容器中，Spring Boot 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。

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。

自动配置

Spring Boot 能根据当前类路径下的类、jar 包来自动配置 bean，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。

无代码生成和XML配置

Spring Boot 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。

应用监控

Spring Boot 提供一系列端点可以监控服务及应用，做健康检测。

缺点

Spring Boot 虽然上手很容易，但如果你不了解其核心技术及流程，所以一旦遇到问题就很棘手，而且现在的解决方案也不是很多，需要一个完善的过程。

3、SpringBoot、Spring MVC和Spring有什么区别？

Spring

Spring最重要的特征是依赖注入。所有Spring Modules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，可以开发松耦合应用。

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。

SpringBoot

Spring和Spring MVC的问题在于需要配置大量的参数。

SpringBoot通过一个自动配置和启动的项来解决这个问题。

4、什么是Spring Boot Starter？

启动器是一套方便的依赖描述符，它可以放在自己的程序中。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粘贴的负载。

例如，如果想使用Spring和JPA访问数据库，只需要项目中包含spring-boot-starter-data-jpa 依赖项，你就可以正常是用。

5、为什么需要spring-boot-maven-plugin？

spring-boot-maven-plugin提供了一些像jar一样打包或者运行应用程序的命令。

spring-boot:run 运行SpringBoot应用程序；

spring-boot:repackage 重新打包你的jar包或者是war包使其可执行

spring-boot:start和spring-boot:stop管理Spring Boot应用程序的生命周期

spring-boot:build-info生成执行器可以使用的构造信息

6、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。

7、SpringBoot自动配置的原理

在Spring程序main方法中，添加@SpringBootApplication或者@EnableAutoConfiguration会自动去maven中读取每个starter中的spring.factories文件，该文件里配置了所有需要被创建的Spring容器中的bean

8、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或者其他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RequestMapping的一个延伸，目的是为了提高清晰度。

##9、spring-boot-starter-parent有什么作用？

我们知道，新建一个SpringBoot项目，默认都是有parent的，这个parent就是spring-boot-starter-parent，spring-boot-starter-parent主要有如下作用：

定义了Java编译版本为1.8

使用UTF-8格式编码

继承自spring-boor-dependencies，这里面定义了依赖的版本，也正是因为继承了这个依赖，所以我们在写依赖时才不需要写版本号

执行打包操作的配置

自动化的资源过滤

自动化的插件配置

针对application.peoperties和application.yuml的资源过滤，包括通过profile定义的不同环境的配置文件，例如application-dev.properties和application-dev.yuml。

10、SpringBoot 打成jar和普通的jar有什么区别？

Spring Boot 项目最终打包成的 jar 是可执行 jar ，这种 jar 可以直接通过java -jar xxx.jar命令来运行，这种 jar 不可以作为普通的 jar 被其他项目依赖，即使依赖了也无法使用其中的类。

Spring Boot 的 jar 无法被其他项目依赖，主要还是他和普通 jar 的结构不同。普通的 jar 包，解压后直接就是包名，包里就是我们的代码，而 Spring Boot 打包成的可执行 jar 解压后，在 \BOOT-INF\classes目录下才是我们的代码，因此无法被直接引用。如果非要引用，可以在 pom.xml 文件中增加配置，将 Spring Boot 项目打包成两个 jar ，一个可执行，一个可引用。

11、运行SpringBoot有几种方式？

打包用命令或者放到容器中运行

用Maven或Gradle插件运行

直接执行main方法运行

12、开启Spring Boot特性有哪几种方式？

继承spring-boot-starter-parent项目

导入spring-boot-dependencies项目依赖

13、什么是Spring Data？

Spring Data 是 Spring 的一个子项目。用于简化数据库访问，支持NoSQL 和 关系数据存储。其主要目标是使数据库的访问变得方便快捷。Spring Data 具有如下特点：

SpringData 项目支持 NoSQL 存储：

MongoDB （文档数据库）

Neo4j（图形数据库）

Redis（键/值存储）

Hbase（列族数据库）

SpringData 项目所支持的关系数据存储技术：

JDBC

JPA

Spring Data Jpa 致力于减少数据访问层 (DAO) 的开发量. 开发者唯一要做的，就是声明持久层的接口，其他都交给 Spring Data JPA 来帮你完成！Spring Data JPA 通过规范方法的名字，根据符合规范的名字来确定方法需要实现什么样的逻辑。

14、什么是Swagger？你用Spring Boot实现了吗？

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。

15、前后端分离，如何维护接口文档？

前后端分离开发日益流行，大部分情况下，我们都是通过 Spring Boot 做前后端分离开发，前后端分离一定会有接口文档，不然会前后端会深深陷入到扯皮中。一个比较笨的方法就是使用 word 或者 md 来维护接口文档，但是效率太低，接口一变，所有人手上的文档都得变。在 Spring Boot 中，这个问题常见的解决方案是 Swagger ，使用 Swagger 我们可以快速生成一个接口文档网站，接口一旦发生变化，文档就会自动更新，所有开发工程师访问这一个在线网站就可以获取到最新的接口文档，非常方便。

16、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。通过实现一个ControlerAdvice类，来处理控制类抛出的所有异常。

17、什么是FreeMarker模板？

FreeMarker 是一个基于 Java 的模板引擎，最初专注于使用 MVC 软件架构进行动态网页生成。使用 Freemarker 的主要优点是表示层和业务层的完全分离。程序员可以处理应用程序代码，而设计人员可以处理 html 页面设计。最后使用freemarker 可以将这些结合起来，给出最终的输出页面。

18、如何实现Spring Boot应用程序的安全性？

为了实现Spring Boot的安全性，使用spring-boot-starter-security依赖项，并且必须添加安全配置。它只需要很少代码。配置类将必须扩展WebSecurityConfigurerAdapter并覆盖其方法。

##19、比较一下Spring Security和Shiro各自的优缺点？

由于Spring Boot官方提供了大量的非常方便的开箱即用的Starter，包括Spring Security的Starter，使得在SpringBoot中使用Spring Security变得更加容易，甚至只需要添加一个一来就可以保护所有接口，所以如果是SpringBoot项目，一般选择Spring Security。当然这只是一个建议的组合，单纯从技术上来说，无论怎么组合，都是没有问题的。

Shiro和Spring Security相比，主要有如下特点：

Spring Security是一个重量级的安全管理框架；Shiro则是一个轻量级的安全管理框架；

Spring Security概念复杂，配置繁琐；Shiro概念简单、配置简单；

Spring Security功能强大；Shiro功能简单

20、Spring Boot中如何解决跨域问题？

跨域可以在前端通过JSONP来解决，但是JSONP只可以发送GET请求，无法发送其他类型的请求，在RESTful风格的应用中，就显得非常鸡肋，因此推荐在后端通过（CORS，Cross-origin resource sharing）来解决跨域问题。这种解决方案并非Spring Boot特有的，在传统的SSM框架中，就可以通过CORS来解决跨域问题，只不过之前我们是在XML文件中配置CORS，现在可以通过实现WebMvcConfigurer接口然后重写addCorsMappings方法解决跨域问题。

项目中前后端分离部署，所以需要解决跨域的问题。

我们使用cookie存放用户登录的信息，在spring拦截器进行权限控制，当权限不符合时，直接返回给用户固定的json结果。

当用户登录以后，正常使用；当用户退出登录状态时或者token过期时，由于拦截器和跨域的顺序有问题，出现了跨域的现象。

我们知道一个http请求，先走filter，到达servlet后才进行拦截器的处理，如果我们把cors放在filter里，就可以优先于权限拦截器执行。

21、什么是CSRF攻击？

CSRF 代表跨站请求伪造。这是一种攻击，迫使最终用户在当前通过身份验证的Web 应用程序上执行不需要的操作。CSRF 攻击专门针对状态改变请求，而不是数据窃取，因为攻击者无法查看对伪造请求的响应。

22、Spring Boot的核心注解是哪些？他主由哪几个注解组成的？

启动类上面的注解是@SpringBootApplication，他也是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；

@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的功能：@SpringBootApplication(exclude={DataSourceAutoConfiguration.class})；

@ComponentScan：Spring组件扫描。

23、SpringBoot的核心配置文件有哪几个？他们的区别是什么？

SpringBoot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/解密的场景

24、SpringBoot有哪几种读取配置的方式？

Spring Boot 可 以 通 过 @PropertySource,@Value,@Environment, @ConfigurationProperties 来绑定变量。

25、Spring Boot 支持哪些日志框架？推荐和默认的日志框架是哪个？

Spring Boot 支持 Java Util Logging, Log4j2, Lockback 作为日志框架，如果你使用Starters 启动器，Spring Boot 将使用 Logback 作为默认日志框架。

26、保护SpringBoot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

27、SpringBoot 2.X有哪些新特性？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

1、SpringBoot自动配置的原理是什么？

SpringBoot启动的时候通过@EnableAutoConfiguration注解找到META-INF/spring.factories配置文件中所有的自动配置类，并对其进行加载，而这些自动配置类的类名都是以AutoConfiguration结尾来命名的，它实际上就是一个javaConfig形式的Spring容器配置类，它们都有一个@EnableConfigurationPerperties的注解，通过这个注解启动XXXProperties命名的类去加载全局配置中的属性，如server.port,而XXXProperties通过@ConfigurationProperties注解将全局配置文件中的属性与自己的属性进行绑定。

2、SpringBoot 配置加载顺序?

1、 properties文件 2、YAML文件 3、系统环境变量 4、命令行参数

3、spring boot初始化环境变量流程?

1、 调用prepareEnvironment方法去设置环境变量

2、 接下来有三个方法getOrCreateEnvironment，configureEnvironment，environmentPrepared

3、 getOrCreateEnvironment去初始化系统环境变量

4、 configureEnvironment去初始化命令行参数

5、 environmentPrepared当广播到来的时候调用onApplicationEnvironmentPreparedEvent方法去使用postProcessEnvironment方法load yml和properties变量

4、运行 SpringBoot 有哪几种方式？

1、 打包用命令或者者放到容器中运行

2、 用 Maven/ Gradle 插件运行

3、 直接执行 main 方法运行

5、SpringBoot 中如何解决跨域问题 ?

跨域可以在前端通过 JSONP 来解决，但是 JSONP 只可以发送 GET 请求，无法发送其他类型的请求，在 RESTful 风格的应用中，就显得非常鸡肋，因此我们推荐在后端通过 （CORS，Cross-origin resource sharing） 来解决跨域问题。这种解决方案并非 SpringBoot 特有的，在传统的 SSM 框架中，就可以通过 CORS 来解决跨域问题，只不过之前我们是在 XML 文件中配置 CORS ，现在可以通过实现WebMvcConfigurer接口然后重写addCorsMappings方法解决跨域问题。

@Configuration

public class CorsConfig implements WebMvcConfigurer {

}

6、SpringBoot如何配置log4j？

在引用log4j之前，需要先排除项目创建时候带的日志，因为那个是Logback，然后再引入log4j的依赖，引入依赖之后，去src/main/resources目录下的log4j-spring.properties配置文件，就可以开始对应用的日志进行配置使用。

7、SpringBoot运行项目的几种方式？

打包用命令或者放到容器中运行

1、 打成jar包，使用java -jar xxx.jar运行

2、 打成war包，放到tomcat里面运行

直接用maven插件运行 maven spring-boot：run

直接执行main方法运行

8、什么是JavaConfig？

Spring JavaConfig是Spring社区的产品，它提供了配置Spring IoC容器的纯Java方法。因此它有助于避免使用XML配置。使用JavaConfig的优点在于：

面向对象的配置。由于配置被定义为JavaConfig中的类，因此用户可以充分利用Java中的面向对象功能。一个配置类可以继承另一个，重写它的@Bean方法等。

减少或消除XML配置。基于依赖注入原则的外化配置的好处已被证明。但是，许多开发人员不希望在XML和Java之间来回切换。

JavaConfig为开发人员提供了一种纯Java方法来配置与XML配置概念相似的Spring容器。

从技术角度来讲，只使用JavaConfig配置类来配置容器是可行的，但实际上很多人认为将JavaConfig与XML混合匹配是理想的。

类型安全和重构友好。JavaConfig提供了一种类型安全的方法来配置Spring容器。由于Java 5.0对泛型的支持，现在可以按类型而不是按名称检索bean，不需要任何强制转换或基于字符串的查找

9、运行 SpringBoot 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

10、SpringBoot 常用的 Starter 有哪些？

1、 spring-boot-starter-web ：提供 Spring MVC + 内嵌的 Tomcat 。

2、 spring-boot-starter-data-jpa ：提供 Spring JPA + Hibernate 。

3、 spring-boot-starter-data-Redis ：提供 Redis 。

4、 mybatis-spring-boot-starter ：提供 MyBatis 。