# **SpringBoot常见的经典面试题**

# 1.SpringBoot的核心注解

启动类上面的注解是@SpringBootApplication，它也是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，比如关闭数据源自动配置功能:

@SpringBootApplication(exclude = { DataSourceAutoConfiguration.class })。

@ComponentScan：Spring组件扫描。

# 2.SpringBoot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ControlerAdvice类，来处理控制器类抛出的所有异常。

# 3.spring Boot常用注解

@SpringBootApplication： SpringBootConfiguration配置类、componentScan扫描包、EnableAutoConfiguration导入其他配置类  
@RestController： @ResponseBody和@Controller的作用。  
@Component,@Service,@Controller,@Repository： 将类注入容器。  
@GetMapping、@PostMapping、@PutMapping、@DeleteMapping： 映射请求，只能接收的对应的请求。  
@AutoWired： 按照类型匹配注入。  
@Qualifier： 和AutoWired联合使用，在按照类型匹配的基础上，在按照名称匹配。  
@Resource： 按照名称匹配依赖注入。  
@Bean： 用于将方法返回值对象放入容器。  
@RequestParam： 获取查询参数。即url?name=这种形式  
@RequestBody： 该注解用于获取请求体数据（body）,get没有请求体，故而一般用于post请求。@PathVariable： 获取路径参数。即url/{id}这种形式。  
@Value： 将外部的值动态注入到 Bean 中。  
    @Value(“${}”)：可以获取配置文件的值。  
    @Value(“#{}”)：表示SpEl(Spring Expression Language是Spring表达式语言，可以在运行时查询和操作数据。)表达式通常用来获取 bean 的属性，或者调用 bean 的某个方法。

# 4.springboot中常用的start

spring-boot-starter-parent 继承父类

mybatis-spring-boot-starter 集成mybatis框架

spring-boot-starter-test:测试模块集成

spring-boot-starter-web:web项目

# 5.在启动的时候运行一些特定的代码

实现接口 **ApplicationRunner** 或者 **CommandLineRunner**

# 6、SpringBoot配置 Undertow 容器

## 6.1、添加 Maven 依赖

<dependency>

<groupId>org.springframework.boot</groupId>

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

<!-- 移除掉默认支持的 Tomcat -->

<exclusions>

<exclusion>

<groupId>org.springframework.boot</groupId>

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

</exclusion>

</exclusions>

</dependency>

<!-- 添加 Undertow 容器 -->

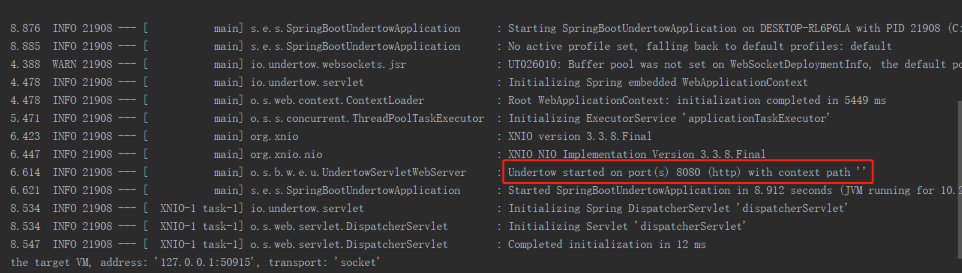
<dependency>

<groupId>org.springframework.boot</groupId>

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

</dependency>

## 6.2、启动项目



## 6.3、Undertow 相关配置

您可以针对 Undertow 容器做一些特定配置，如日志输出路径，设置工作者线程的个数等参数优化等，如下所示：

# 是否打开 undertow 日志，默认为 false

server.undertow.accesslog.enabled=false

# 设置访问日志所在目录

server.undertow.accesslog.dir=logs

# 指定工作者线程的 I/0 线程数，默认为 2 或者 CPU 的个数

server.undertow.io-threads=

# 指定工作者线程个数，默认为 I/O 线程个数的 8 倍

server.undertow.worker-threads=

# 设置 HTTP POST 内容的最大长度，默认不做限制

server.undertow.max-http-post-size=0

# 7、自定义Start



## 7.1、这个场景需要依赖什么

## 7.2、[如何编写自动配置](https://note.clboy.cn/" \l "/backend/springboot/customizestarter?id=%e5%a6%82%e4%bd%95%e7%bc%96%e5%86%99%e8%87%aa%e5%8a%a8%e9%85%8d%e7%bd%ae)

## 7.3、模式

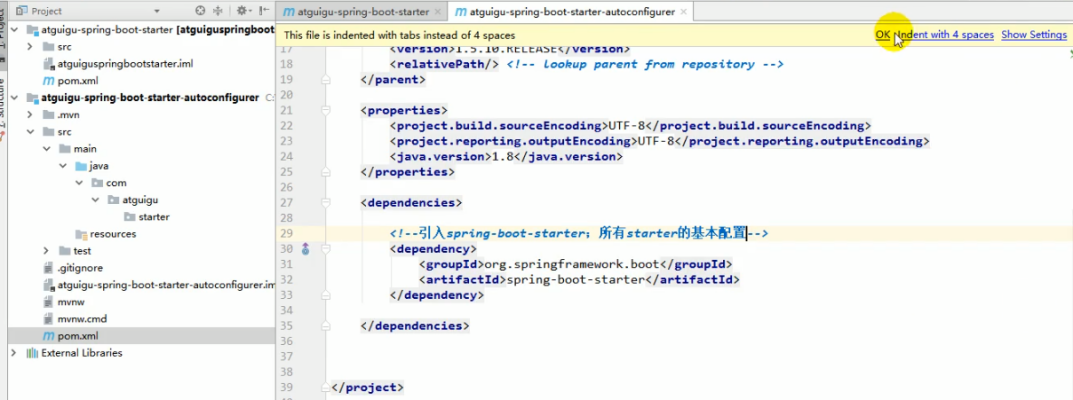


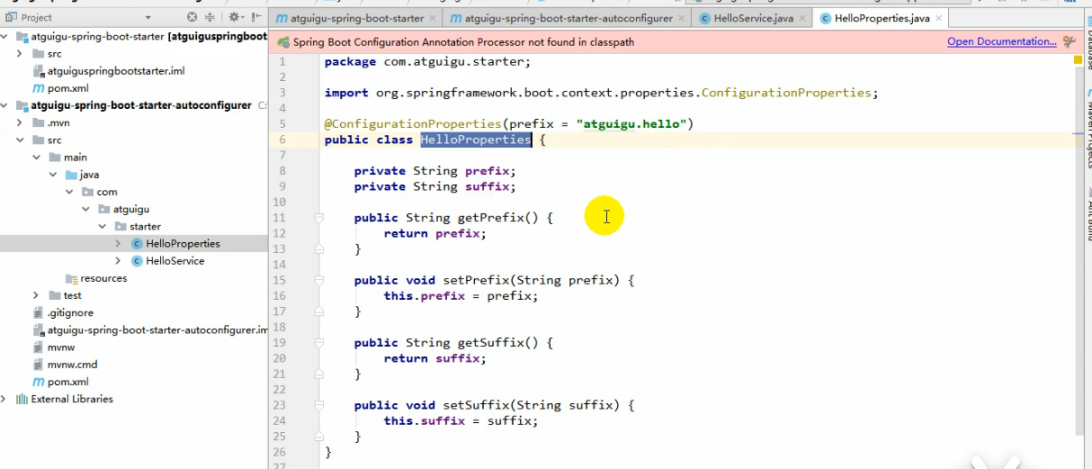
### start模块

自定义的starter使用xxx-spring-boot-starter命名规则



### 自动配置模块



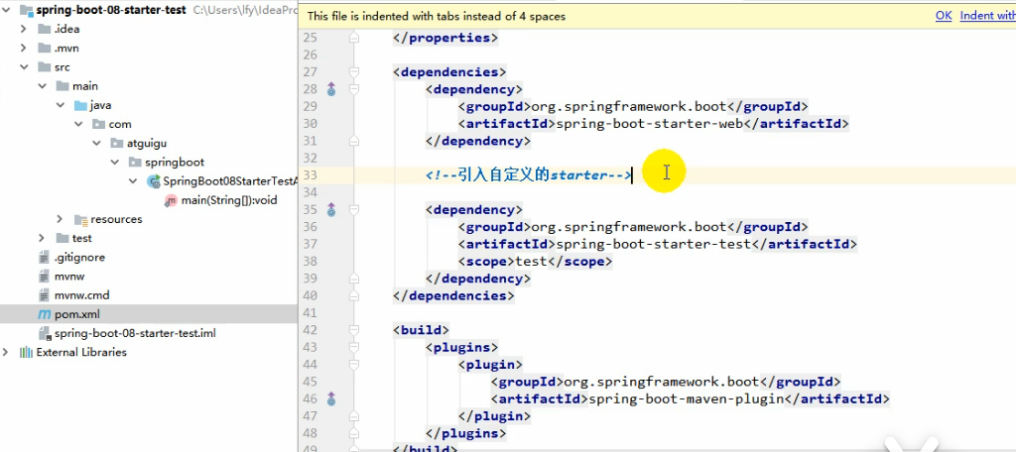


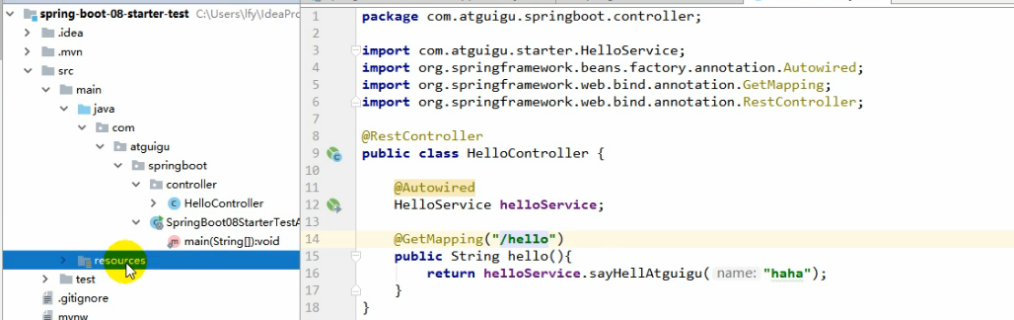


resources下创建META-INF/spring.factories，在该文件中配置自己的自动配置类。



### 测试





# **8、spring中resource和autowire的区别**



