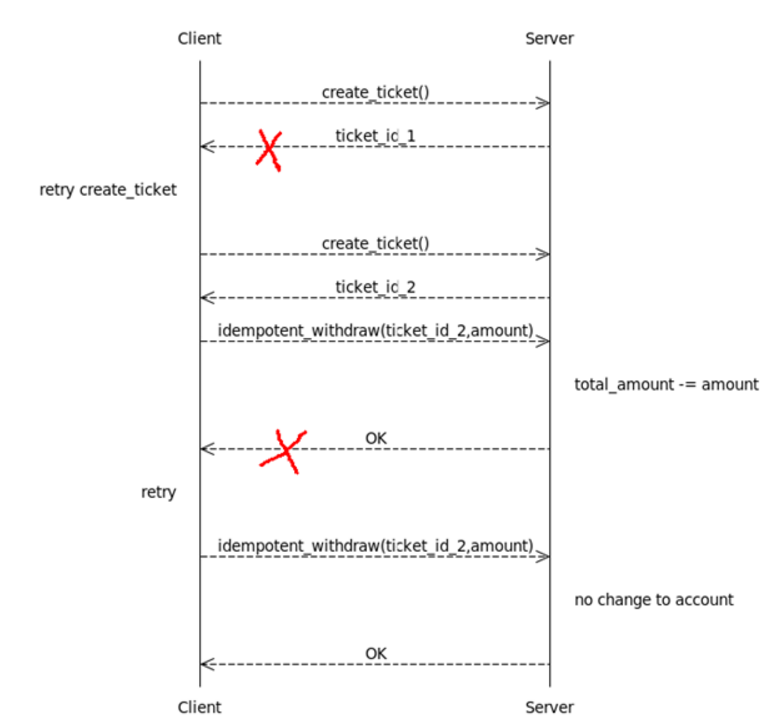
## HTTP幂等性

HTTP方法的幂等性是指一次和多次请求某一个资源应该具有同样的副作用。即每次请求的结果是一样的，GET表示获取操作，是幂等的；PUSH是更新操作，是幂等的(更新后，多次提交结果任然不变，即更新后的结果)；DELETE是删除操作，和PSUH类似；POST保存操作，多次提交，多次保存，非幂等。

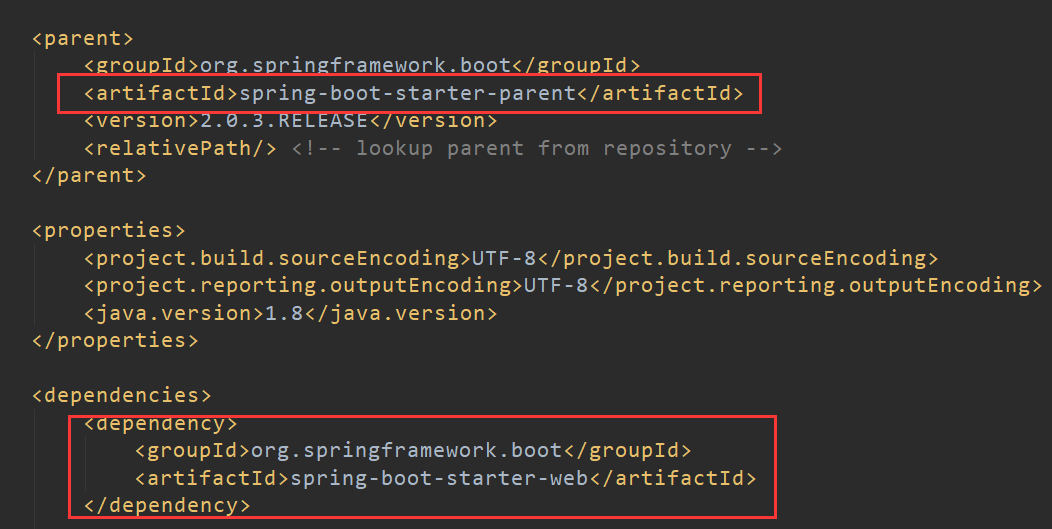
非幂等实现幂等的操作：在扣钱操作中，服务器端扣款成功，但是在将扣款成功返回给客户端时，丢包，客户端认为操作未成功，重复提交，这样就会导致多次扣款，非幂等；在扣款操作前，客户端首先发送一个请求获取一个id，获取到id后，执行扣款操作(会将此id同时发送)服务器端扣款成功，但是返回结果时丢包，客户端可以操作失败，重新发起扣款请求(带上id)，服务器端接收到请求后，发出此id以扣款，且成功，服务器端返回操作成功给客户端



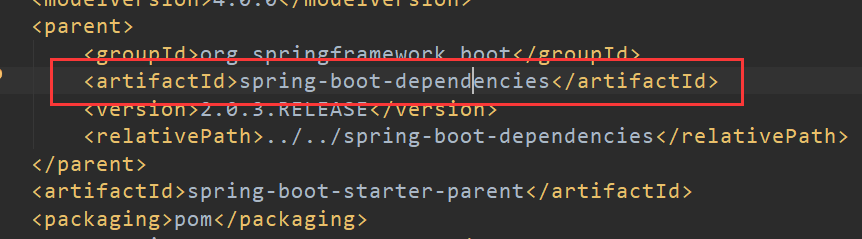
@Controller，@Service，@Respository注解都是@Component注解的子注解

@RequestParam实现自动类型转换

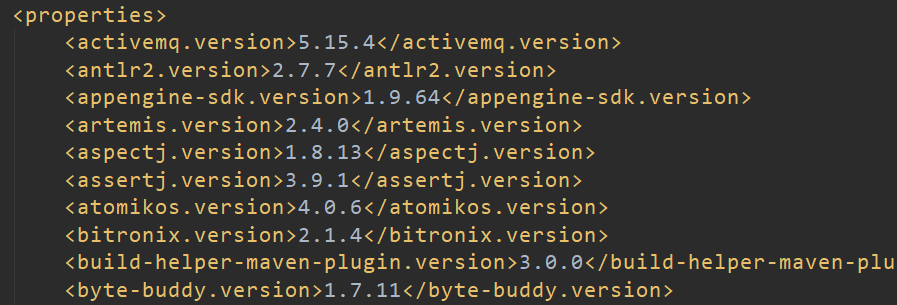
在SpringBoot中在引入依赖的时候，为什么不需要定义依赖的版本(大部分)



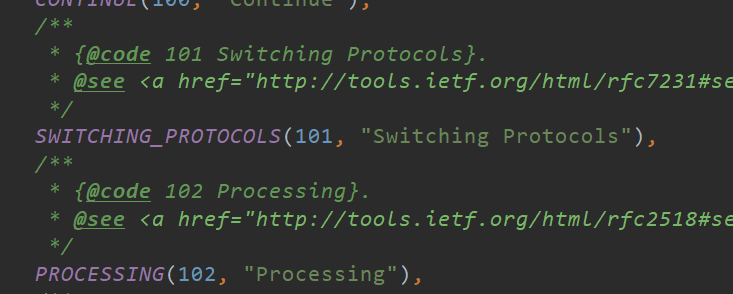
进入spring-boot-parent中



进入spring-boot-dependencies下



当出错时，会显示响应码，可以在HttpStatus中查找对应的响应码及其错误原因



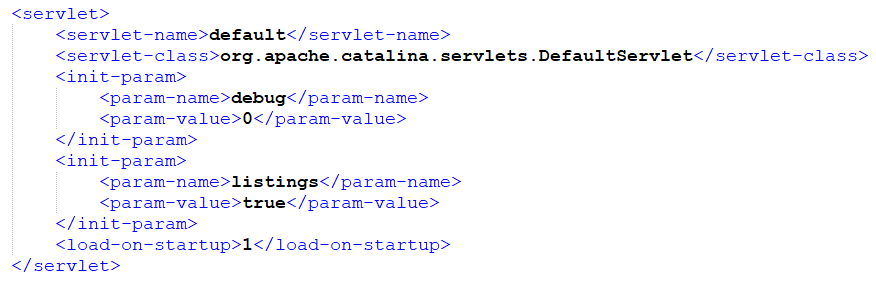
@SpringBootApplication注解实现了@Configuration，@ComponentScan，@EnableAutoConfiguration

@Configuration注解主要和@Bean结合使用

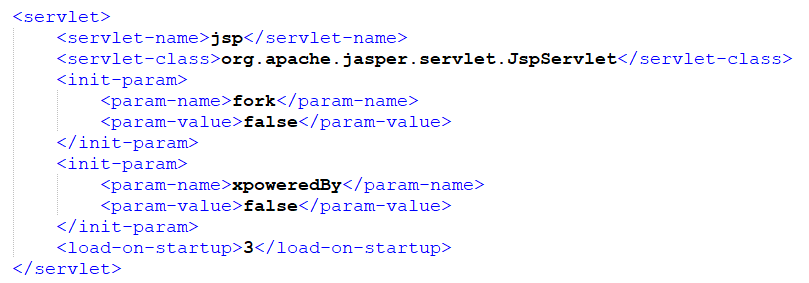
@ComponentScan注解用于指定扫描的路径，SpringBoot默认会扫描main下类及其子包下的类

@EnableAutoConfiguration注解会在你开启某些功能的时候自动配置，这个注解会告诉SpringBoot根据添加的jar依赖猜测你想如何配置Spring，比如导入spring-boot-starter-web时，这可能会是一个web项目，SpringBoot就会自动配置web相关

Tomcat：在tomcat的web.xml文件中声明了两个Servlet，一个DefaultServlet，一个JspServlet，



DefaultServlet的作用是处理静态资源，英文描述时，当其他其他请求为映射到Servlet时，执行此Servlet进行静态资源的查询，这里的listings表示是否罗列静态资源文件，默认false



JspServlet的作用就是映射.jsp请求，找到对应的jsp页面，并渲染为Servlet

Tomcat在运行的时候先查询自身的web.xml，再查询自定义的web.xml

### SpringBoot中常用注解及其分析：

@SpringBootApplication：main方法所在类上，主要由以下4个注解组成

@EnableAutoConfiguration：自动装配，会根据导入的包将对象加载到IOC中

@SpringBootConfiguration：和Spring的@Configuration一样，@SpringBootConfiguration就继承了@Configuration注解，@Configuration和@Bean注解一起使用

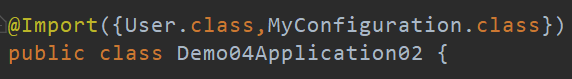
@ComponentScan：去扫描当前类及其类所在的子包的所有类

@Inherited：这是Java的注解，注解A添加@Inherited，在类B添加注解A，类C继承类B，类C上也有注解A，类在继承时是不会继承类的注解的，但是使用@Inherited的注解的类会继承注解

@Configuration/@Bean：

@Configuration用于修饰类，@Bean用于修饰方法，如果通过name获取@Bean的对象，name是方法名，而不是类名

@Import：

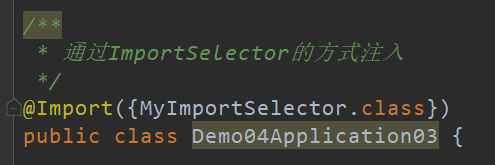


@Import注解的作用是将指定类加入到IOC中

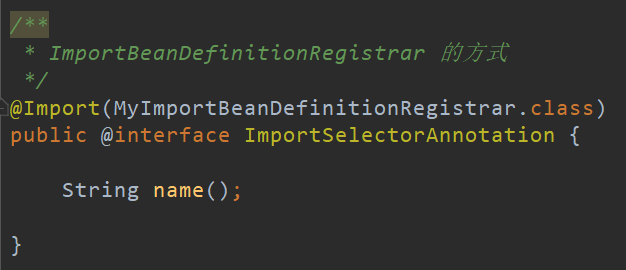
其他实现方式：

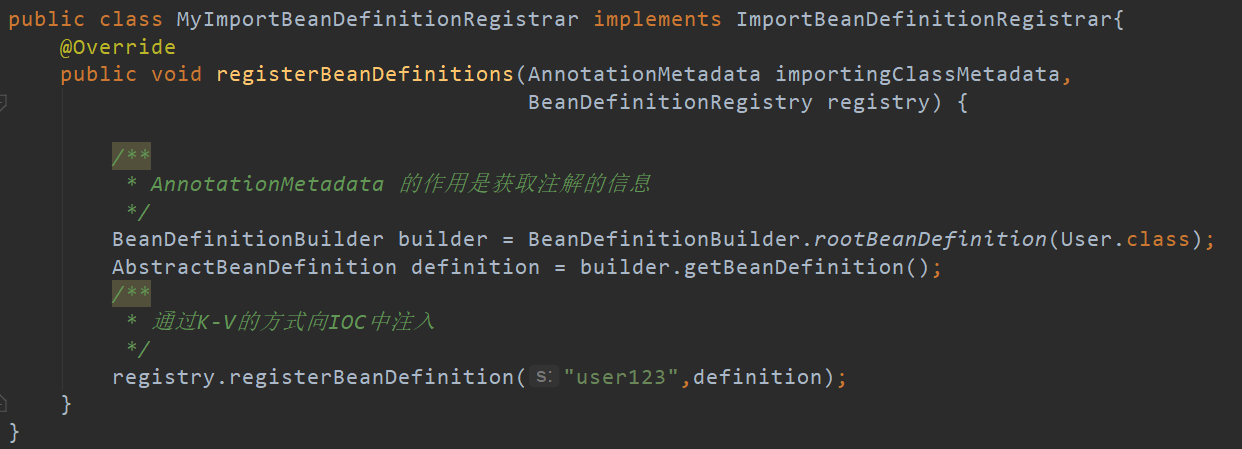


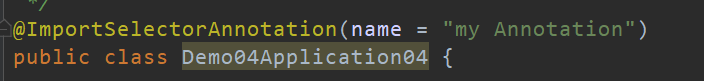
继承ImportSelector，将需要加入到IOC容器中的类的路径作为数组返回，这些类就会被加入到IOC中



自定义注解的方式：

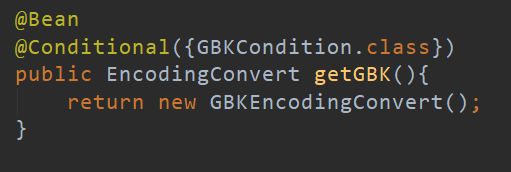


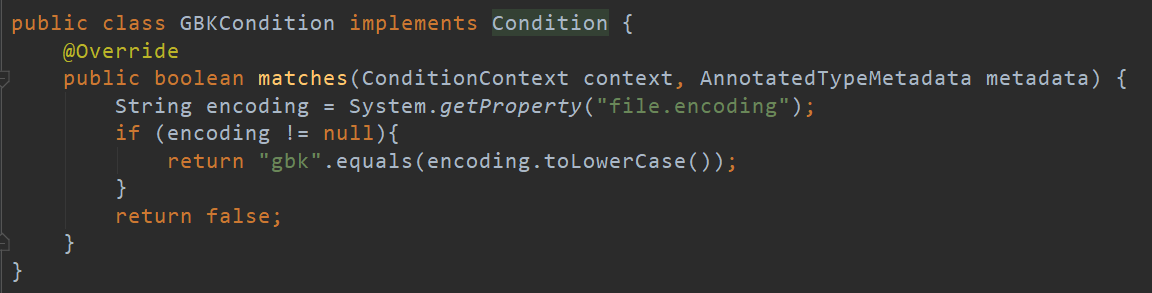




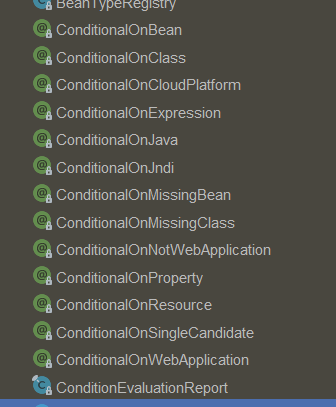
一个是ImportSelector，一个是ImportBeanDefinitionRegister

@Conditional一般和@Bean一起使用，用于判断是否要将当前Bean装载到IOC中



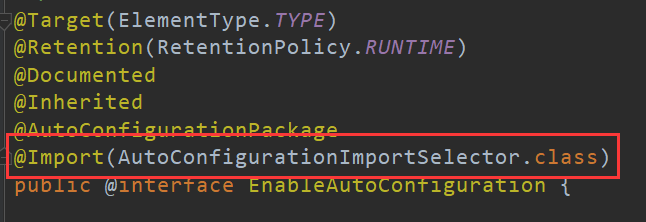


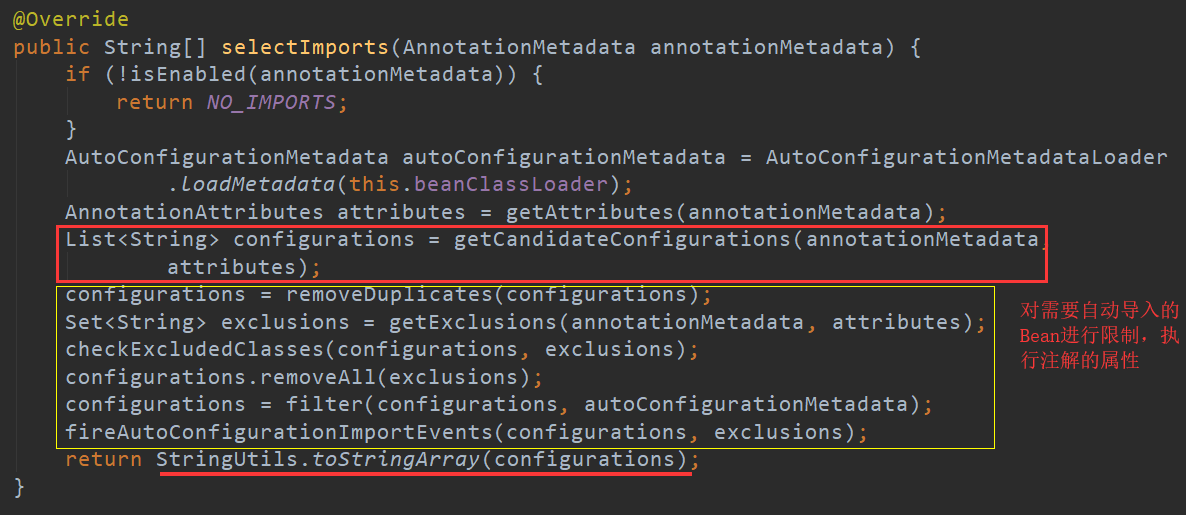
@Conditional传入Class，这个Class必须继承Condition接口，如果接口的matche方法返回true时，才会执行@Bean注解，将Bean装载到IOC中

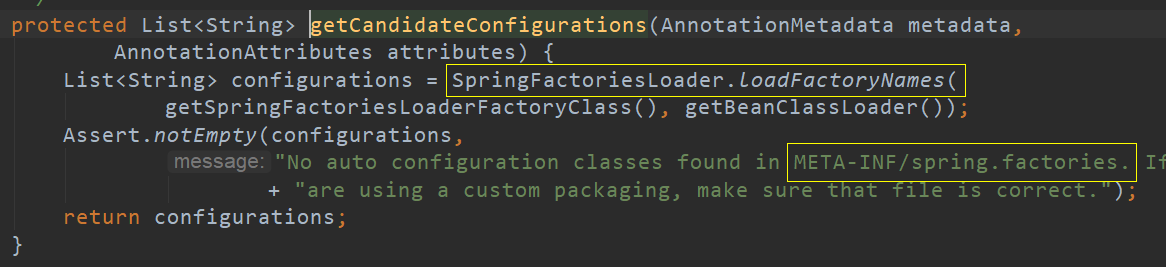


在autoconfigure包下的condition下有很多定义好的@Conditional注解，可以直接使用

@EnableAutoConfiguration自动装配原理：

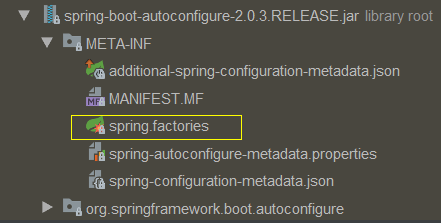




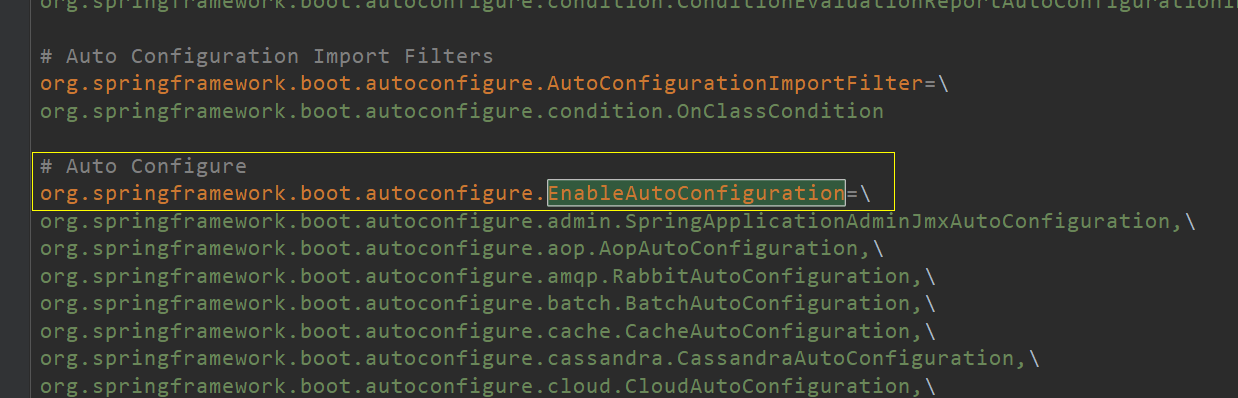


SpringFactoriesLoader.loadFactoryNames会加载ClassPath下的META-INF的spring.factories文件

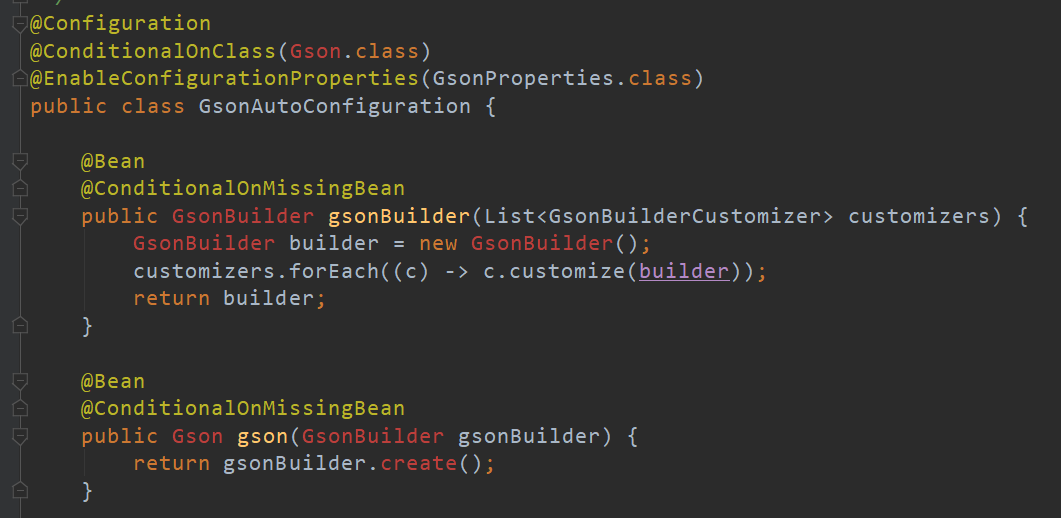
比如，Spring-Boot-autoconfigure.jar下的spring.factories



spring.factories文件，这里的\是换行符



其它的先不管，那么Spring-boot如何实现判断到底要装入哪些Bean



这以GsonAutoConfiguration为例

@Configuration和@Bean就是实例化Bean

@ConditionalOnClass才是关键，表示只有加载了Gson.class，才会执行实例化此Bean，也就是说，如果没有导入Gson相关的包，那么Spring-boot将不会实例化Gson，这也是为什么在导入包后，Spring-boot会自动装配的原因，Spring-Boot会去猜测是否导入了响应的包的关键就是@ConditionalOnClass注解

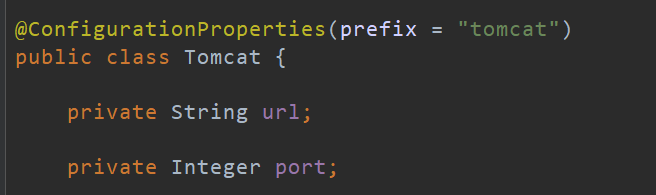
@ConditionalOnMissingBean注解表示，如果IOC中已经有此Bean了，那么不再执行@Bean，确保系统中只有一个Bean

我们也可以在ClassPath下自己创建一个spring.factories文件，以org.springframework.boot.autoconfigure.EnableAutoConfiguration为K，将需要自动装配的Bean的类路径(包含第三方Jar包)写到这里，这样在项目启动的时候，就会自动装配到IOC中

自动装配的判断条件就是根据Conditional注解判断，实现自动装配

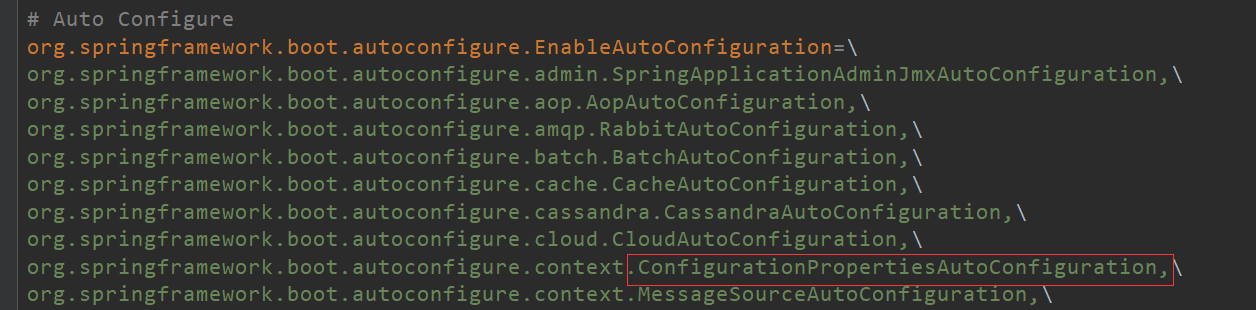
这里的K是org.springframework.boot.autoconfigure.EnableAutoConfiguration，表示@EnableAutoConfiguration注解自动装配的，还有可能其他注解也有类似功能

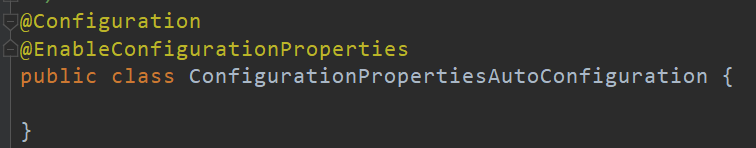
@ConfigurationProperties

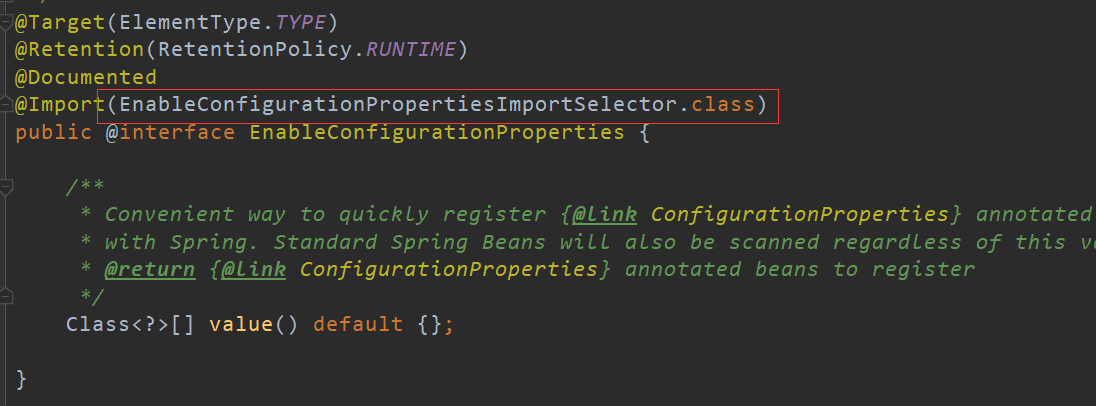


读取application.properties配置文件中的属性，并将Bean对象存放到IOC中

@EnableConfigurationProperties的作用是启动@ConfigurationProperties







@EnableAutoConfiguration在判断导入IOC中的类时，会创建ConfigurationPropertiesAutoConfiguration，此类上有一个@EnableConfigurationProperties，@Import会进实现@ConfigurationProperties类的加入到IOC中，所以就不再需要@Component，同时会实现@ConfigurationProperties的功能，将 application.properties 配置文件的属性装配到Bean中



异常拦截：



在每个类中对异常进行拦截



@ControllerAdvice定义全局异常

SpringBoot的pom文件

在SpringBoot中最主要的几个pom文件：

spring-boot-dependencies：定义一些jar包的版本

spring-boot-starter-parent：定义一些maven，git，idea，eclipse的插件，默认使用JDK1.8，UTF-8编码

spring-boot-dependencies(父)

spring-boot-starter-parent

spring-boot-starter：定义一些SpringBoot的基础(必须的)

spring-boot-starters(父)

spring-boot-parent(父)

spring-boot-dependencies(父)

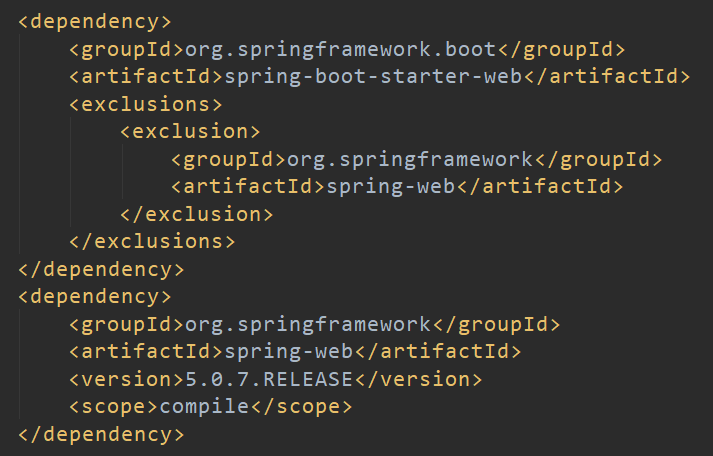
spring-boot-starter-web：web项目需要导入的

spring-boot-starters(父)

spring-boot-parent(父)

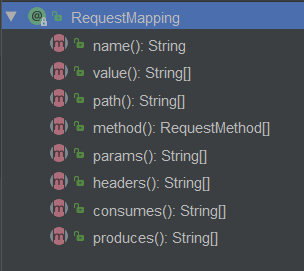
spring-boot-dependencies(父)

spring-boot-starter-web是spring-boot-starters的子pom，但是spring-boot-starter-web引入了spring-boot-starter的依赖，因为这是SpringBoot启动必须的，其他spring-boot-starter-Xxx都是类似的，spring-boot-starter和spring-boot-starters是必须的



Pom文件有多个依赖，可以根据项目的实际需求导入依赖，比如在web中不是使用spring-boot-starter-web自定义的spring-web版本，而使用自定义的5.0.7版本

@ResponseBody注解的作用是将方法的返回值存放到响应的body中



@RequestMapping注解属性详情

Name：实际使用少

Value和path都一样请求的url

Method：请求方式 post，get，put…..

Params：缩小请求范围，params={age!=10&&name==jack}，请求参数中name属性等于jack，并且age属性不等于10才执行方法

Headers：请求头必须包含的

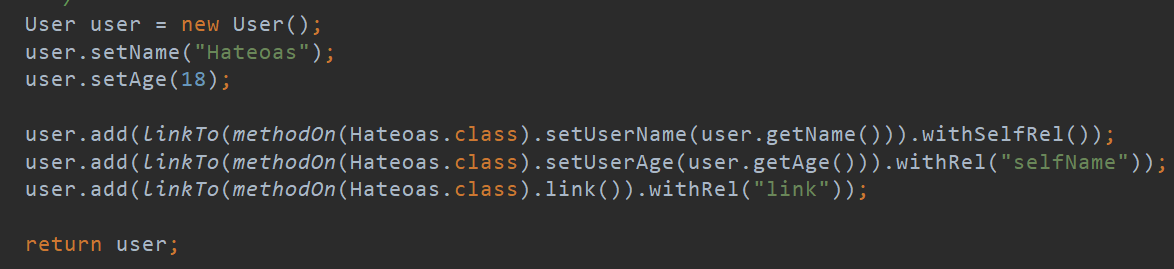
produces：生产者 告诉用于用什么方式解析返回的结果

consumes：消费者 当浏览器能够识别的类型(Accept)和consumes相同时，才访问此方法

hateoas：可以往返回的结果添加Url连接



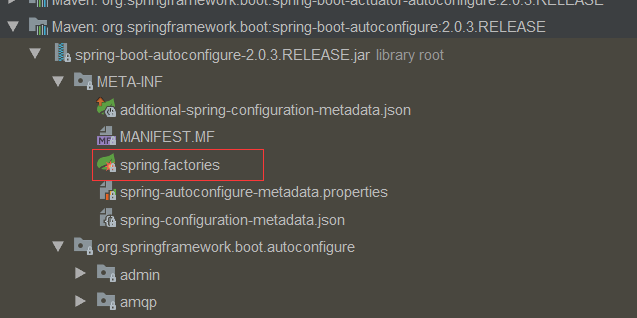
先继承ResourceSupport类，add为ResourceSupport的方法



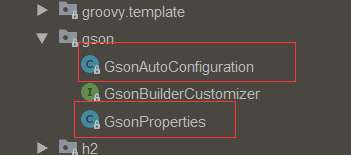
返回结果



autoconfigure包：



自动装配的依据



比如装配GSON相关，GsonAutoConfiguration是否装配的条件

GsonProperties配置Gson相关的参数

