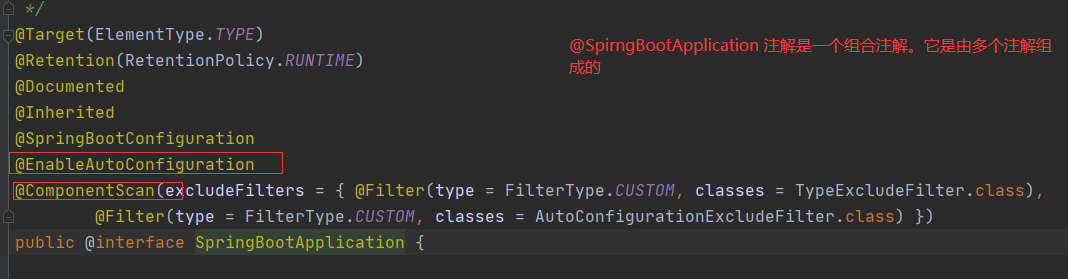
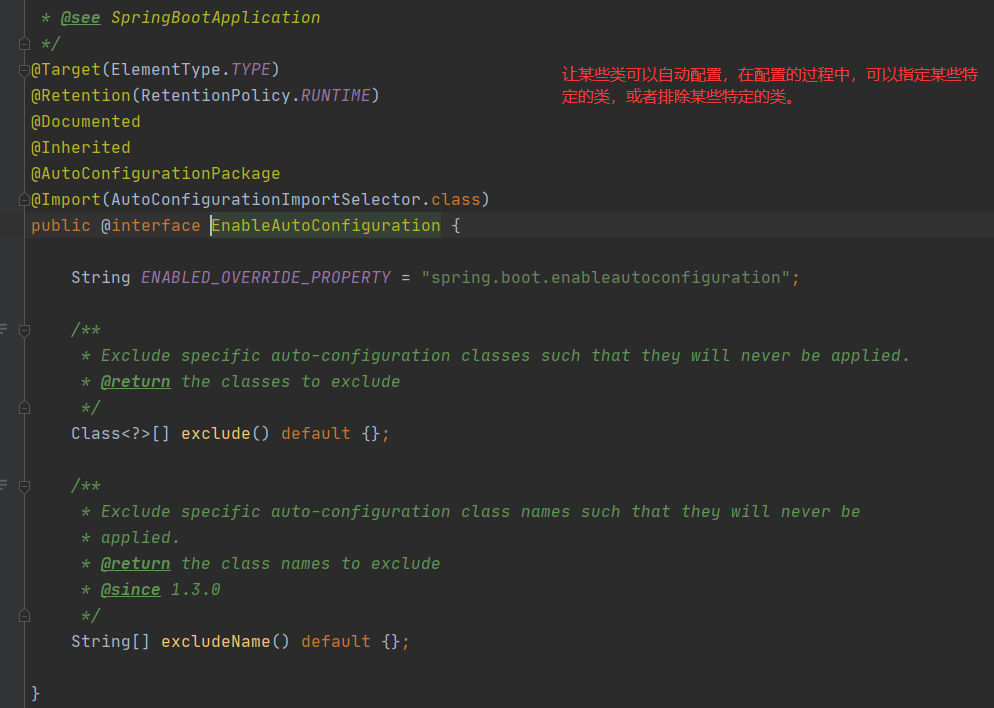
## SpringBootApplication注解详解

@SpringBootApplication注解



@SpringBootApplication 注解是一个组合注解，可以被 @EnableAutoConfiguration 和 @ComponmentScan替换。



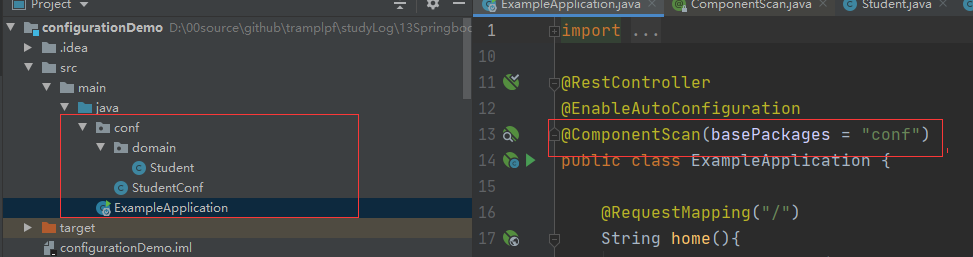


## 2. springBoot 中使用注解类@Configuration

SpringBoot 中使用 @Configuration 注解，可以在项目启动的时候，执行该类，将该类中的方法执行生成相关的bean，然后注入到spring 容器中。

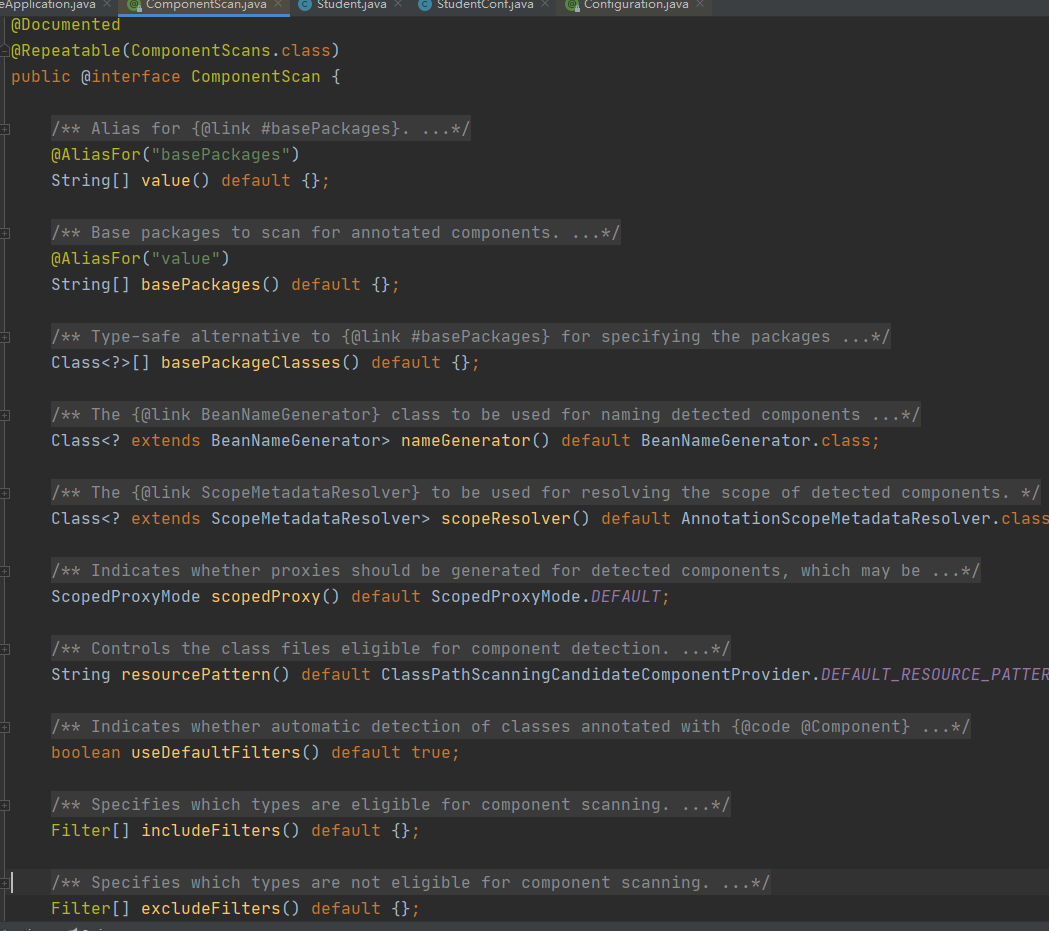
@Configuration 注解是一个组合注解，它包含 @Component， 但是在项目中，如果启动类中，使用的是@EnableAutoConfiguration，而没有使用@SpringBootApplication 或者 没有使用@ConponmentScan，又或者 @ComponmentScan 注解定义的扫描路径不包含被@Configuration 修饰的类，那么这个配置类，在启动的时候，也不会被执行，当然对应的配置类也就不会生效。

只有被@Configuration 修饰的类在 @ComponmentScan 定义的扫描路径下，它才会生效。

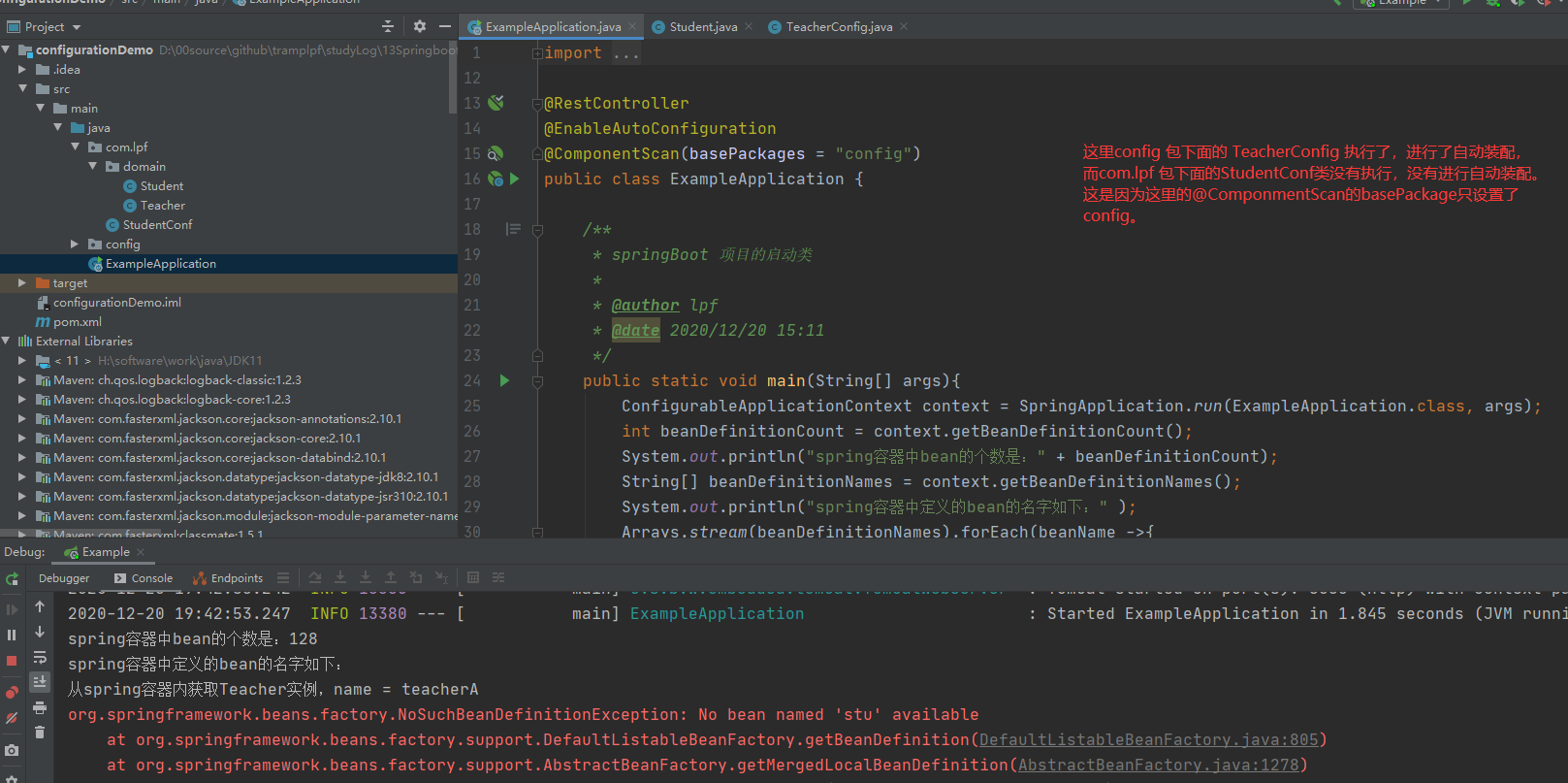


如图：StudentConf 位于conf包下，conf包属于项目的扫描路径，因此它可以被扫描。

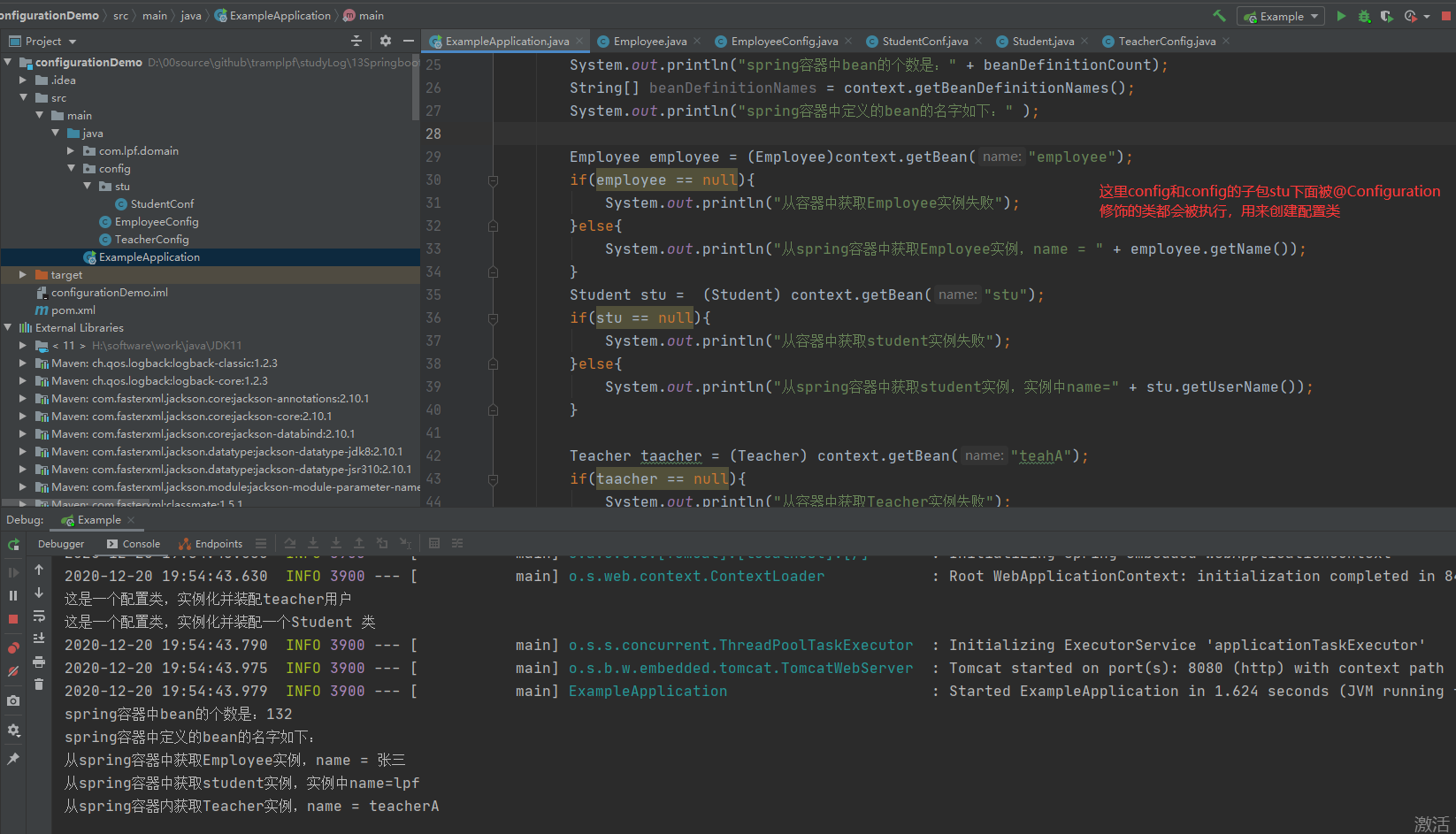
## @ComponmentScan 注解



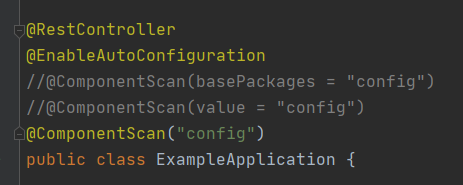
@Configure 配置的类，需要位于 @ComponmentScan的basePackage 下定义的包下面，以及包下面的子包下面。

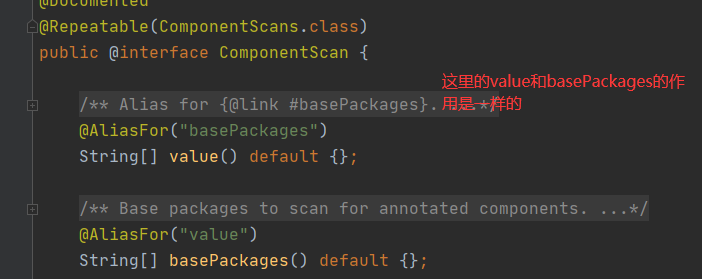


@ComponmentScan() 中basePackage 中定义的包，及其子包下面的类都会被扫描。



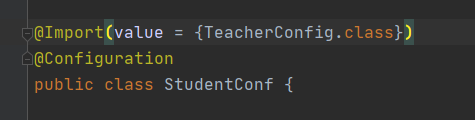
@ComponmentScan 注解中，定义扫描路径的三种方式，如下，下面的三种形式的作用是一样的。





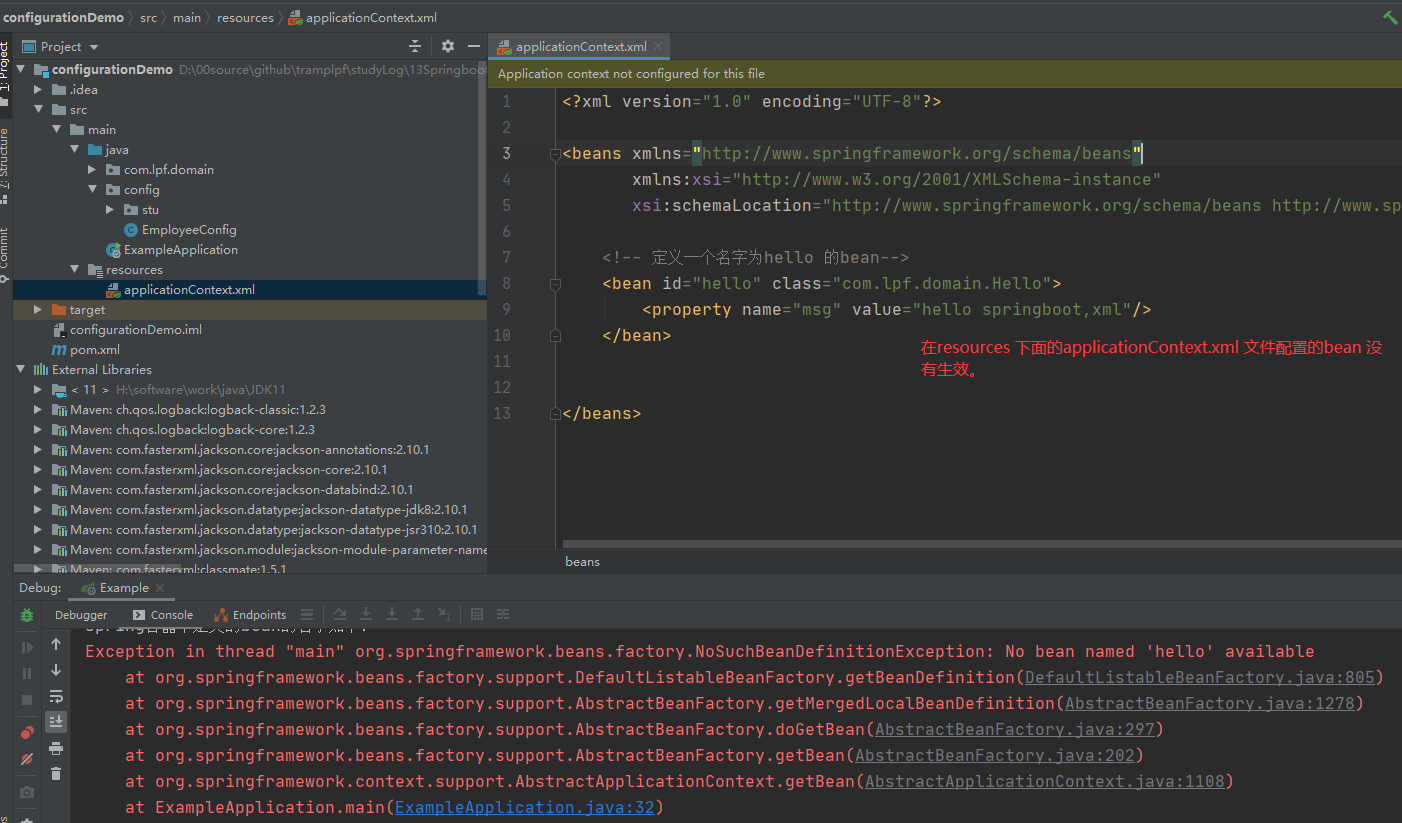
## @Import 注解

<https://www.cnblogs.com/yichunguo/p/12122598.html>

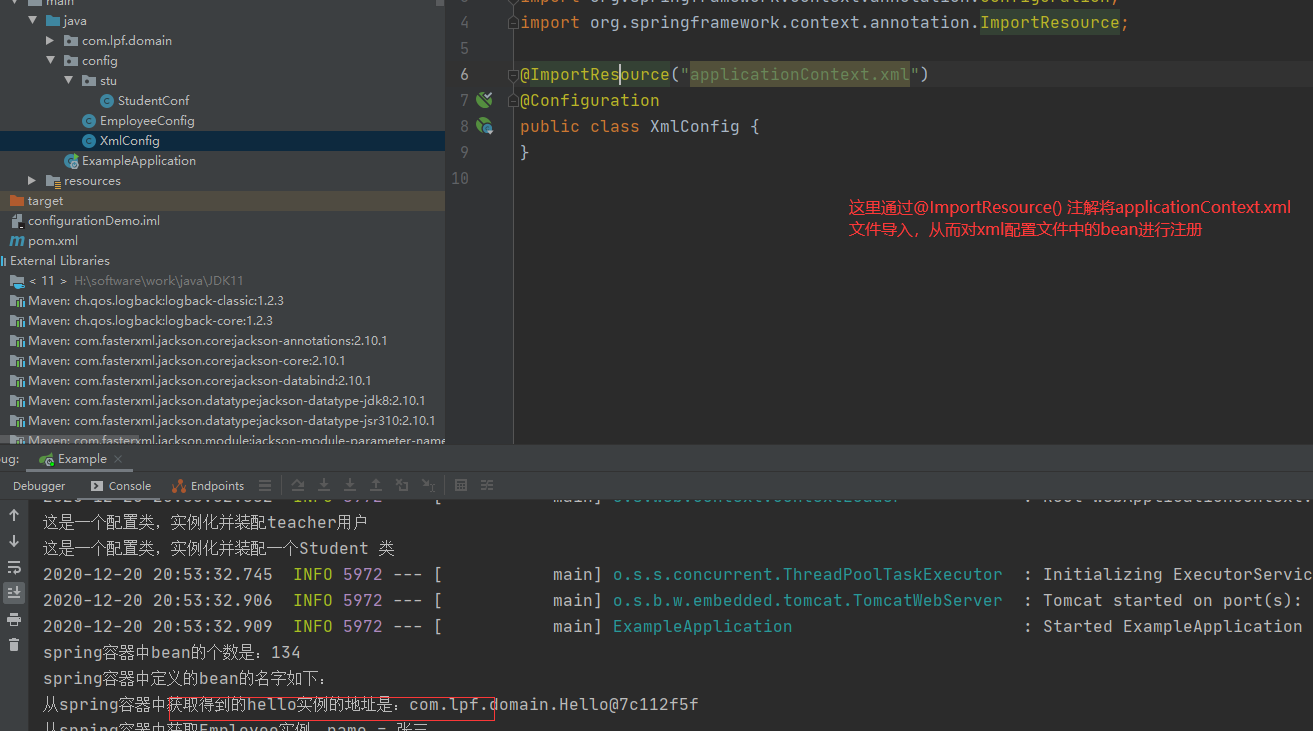


@Import 注解输入的取值中，输入是输入 一个class的数组，通过{} 来输入。

## Springboot 中使用xml配置文件进行配置

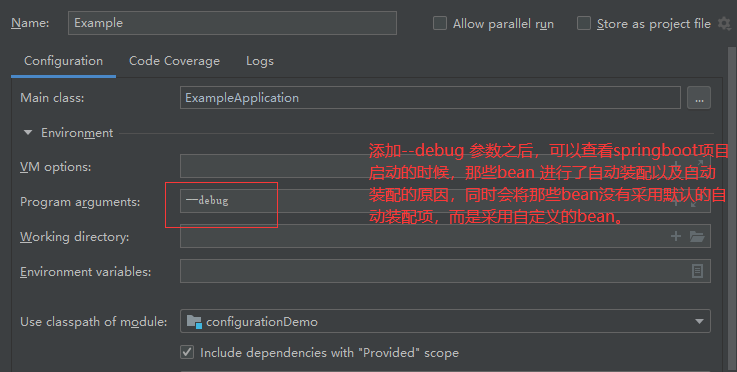


[这里通过@ImportResource(“applicationContext.xml”)](mailto:这里通过@ImportResource() 将配置文件导入到springboot的配置项中。

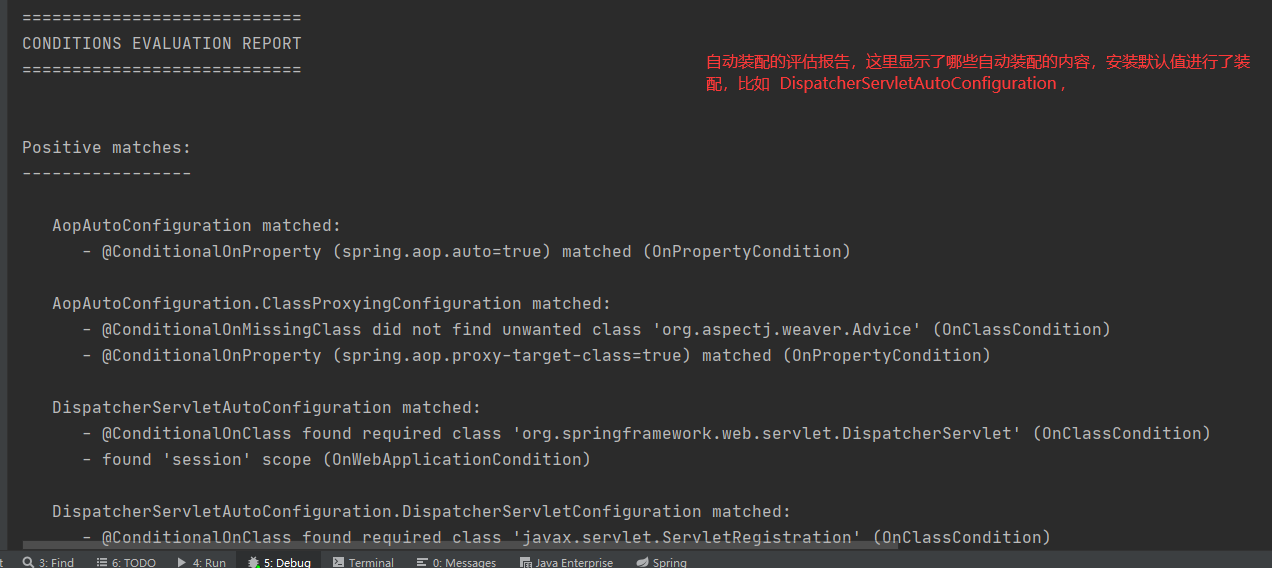


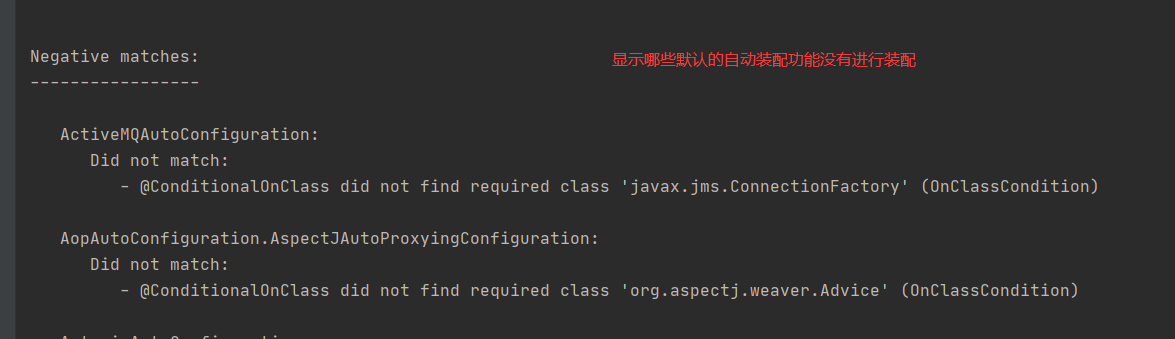
## Springboot 项目查看对那些内容进行了自动装配，以及进行自动装配的原因

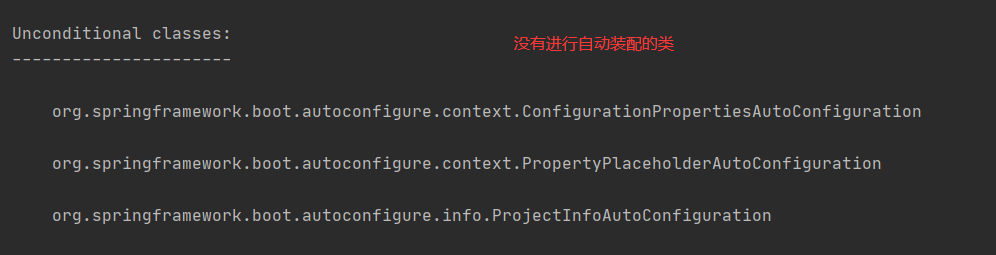
方法： 在项目启动的时候，添加 –debug 参数， 这样会将springboot 将那些bean 进行了自动装配，以及哪些bean 没有进行自动装配进行打印



下面显示哪些进行了自动装配





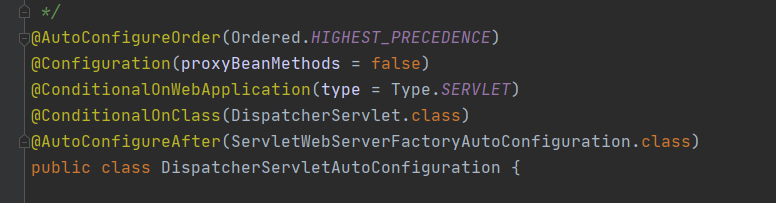


疑惑：

不清楚，这里的Negative matches和 Unconditional classes 之间的区别?

## 针对某一个自动装配项进行解析

这里针对 DispatcherServletAutoConfiguration类进行自动状态。

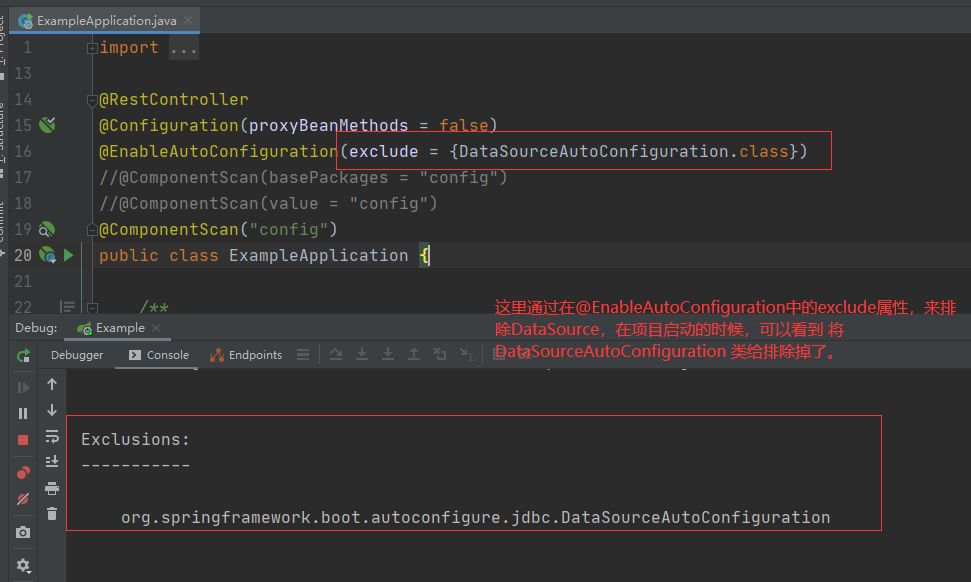


上面的注解，定义了什么情况下对DispatcherServletAutoConfiguration 进行装配(@ConfigurationOnWebApplication， @ConditionalOnClass )

以及自动装配的顺序（这里是要在 ServletWebServetFactoryAutoConfiguration 类装配后进行装配），

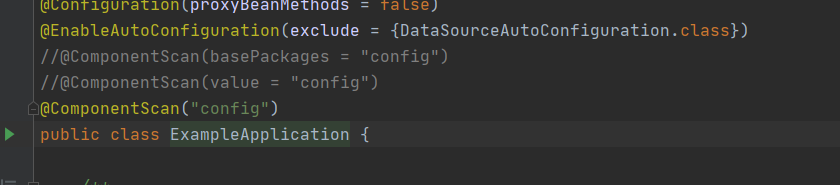
还定义了装配的优先级， @AutoConfigureOrder

定义了装配的方式 @Configuration(proxyBeanMethods = false)



## springBoot排除某些自动配置项

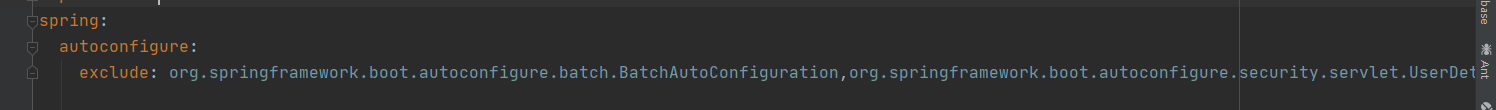
方式一：在@EnableAutoConfiguration 注解中添加 exclude 属性，注意，这里添加的是简单的类名，不需要添加包名



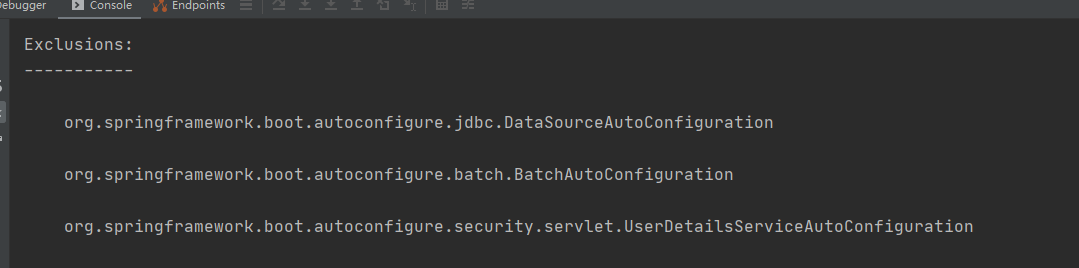
方式二： 在配置文件中，配置spring.autocinfigure.exclude 来指定不需要进行自动装配的类，

注意这里的类是类的全限定类名。

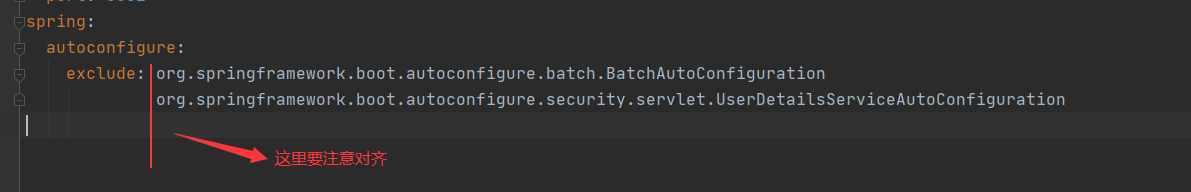
这里要注意，不同的类之间，通过逗号分隔，写在同一行，否则会报错。



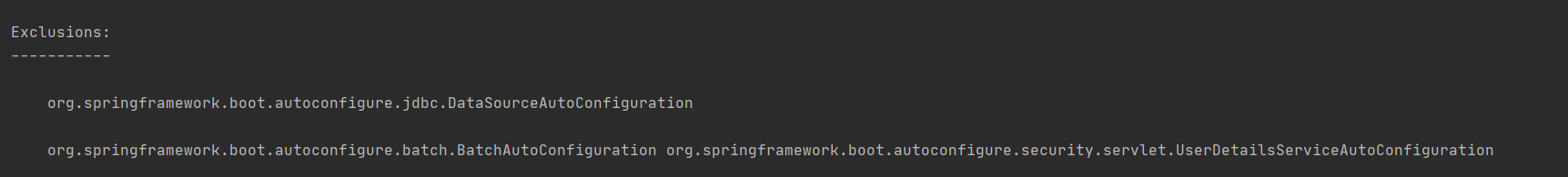
结果如下：



同样，不同的类之间也可以写在不同行，如下图

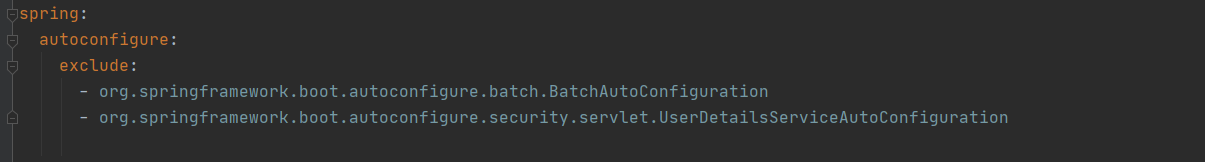


结果如下

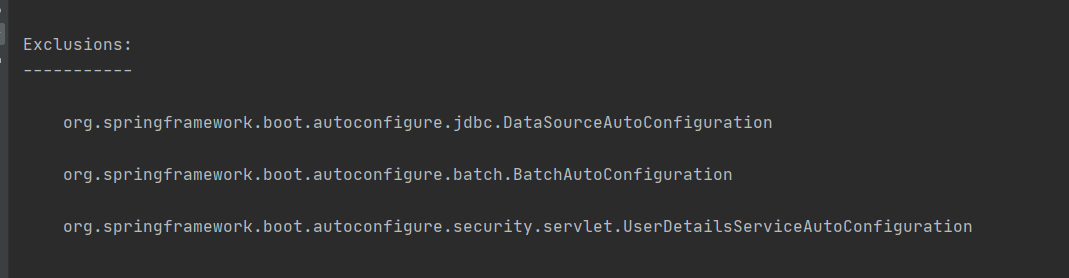


疑惑： 这里将要排除的类写在不同行的时候，为什么显示结果却，和前面的结果不一样。

方式三：可以写成如下的形式，在要排除的类前面，添加 - ，

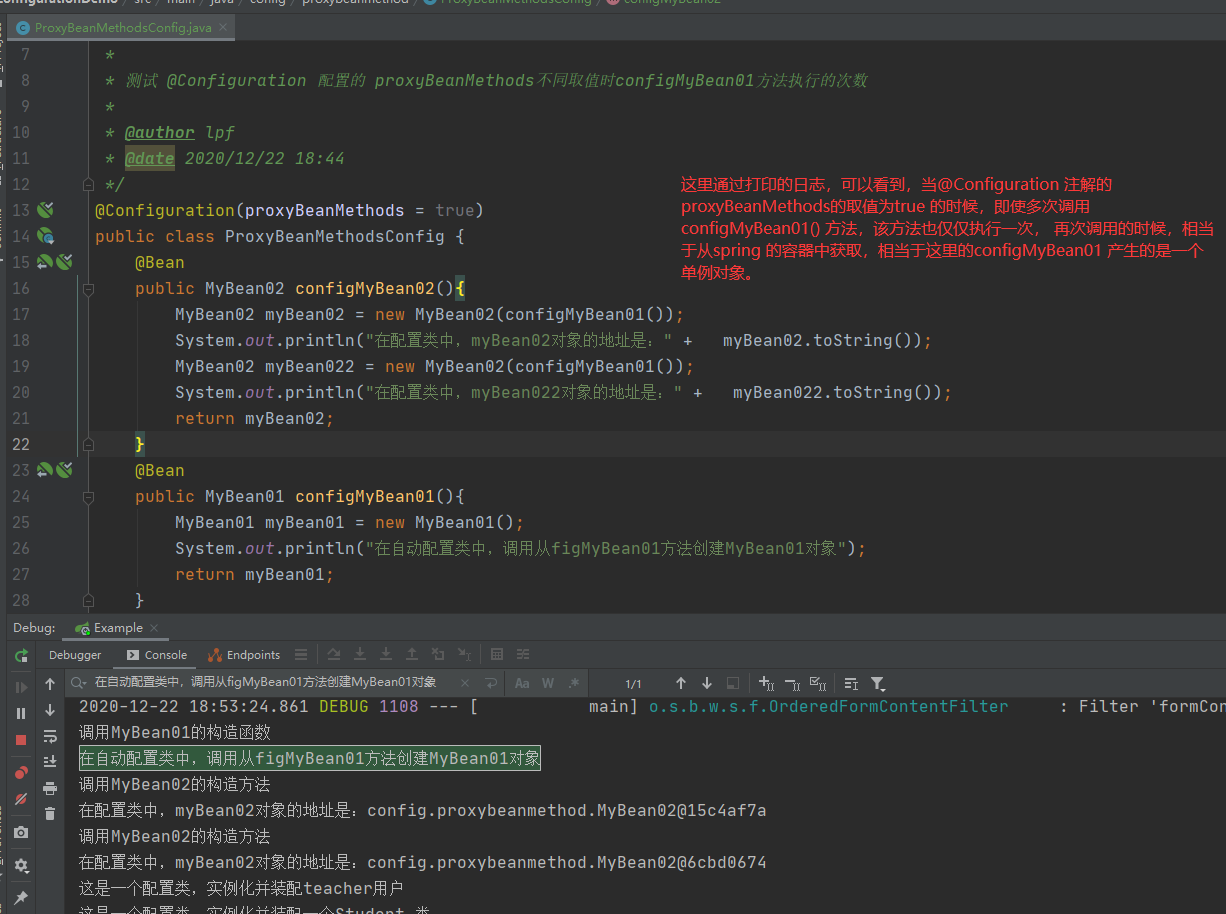


结果如下：

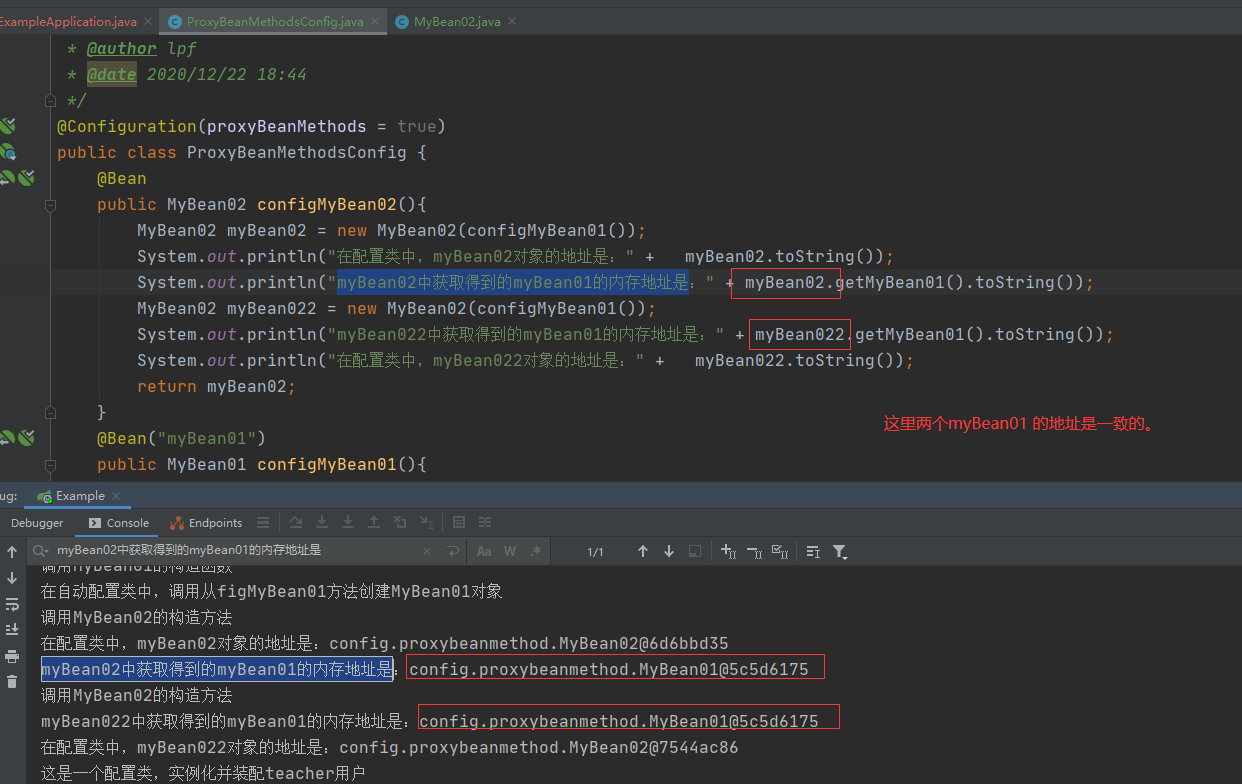


## 远程调试springboot 项目

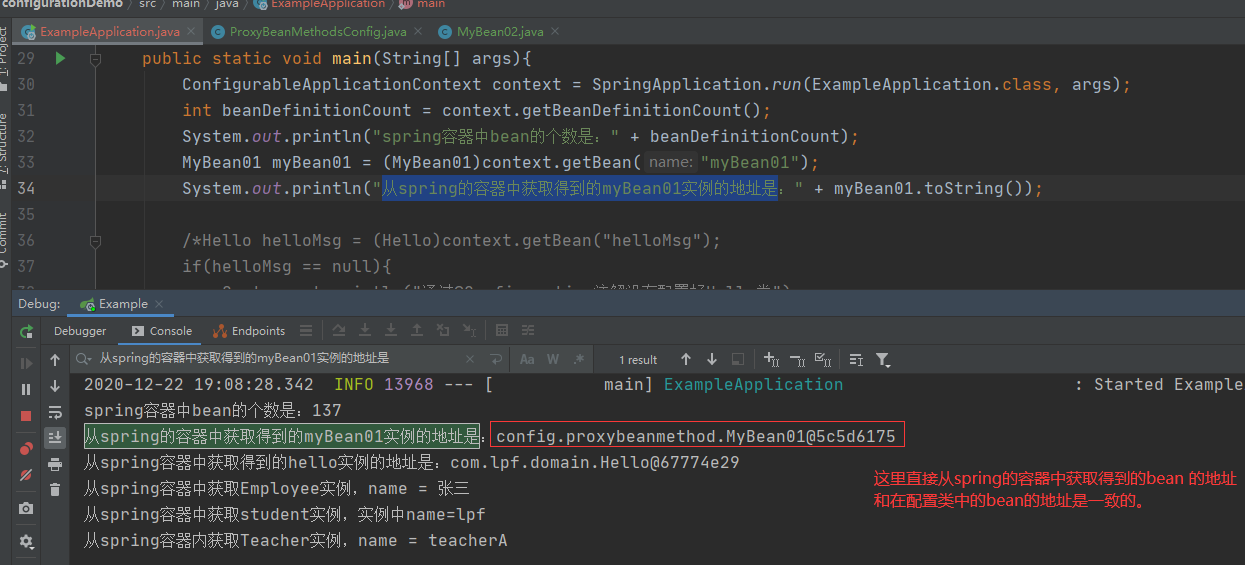
## @Configuration 注解的 proxyBeanMethods 属性



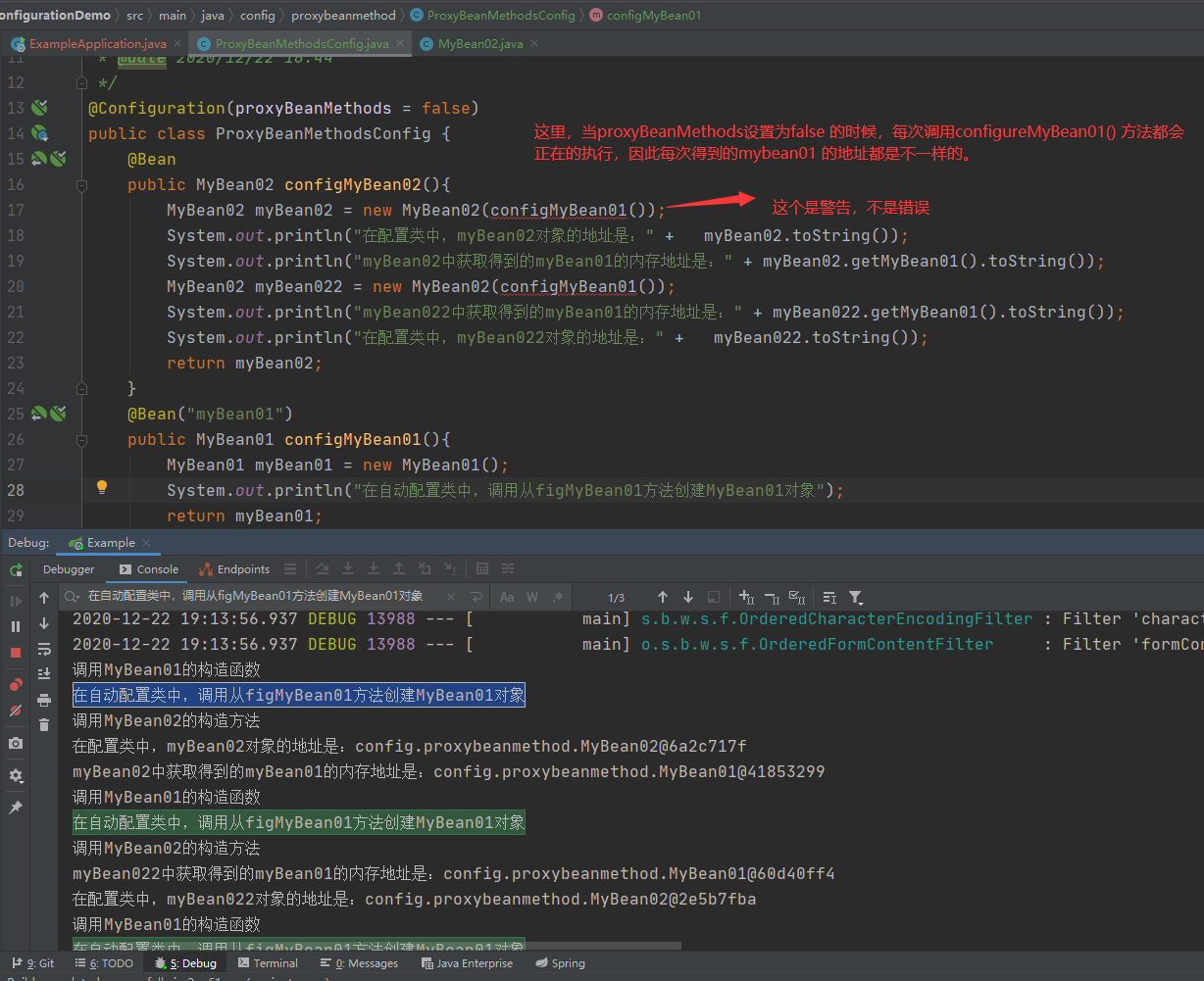
当@Configuration 注解的proxyBeanMethods 的取值为true时候，配置类中被 @Bean 修饰的实例化bean的方法只会执行一次。 第二次调用被@Bean 修饰的方法的时候，会从spring 的容器中获取bean。



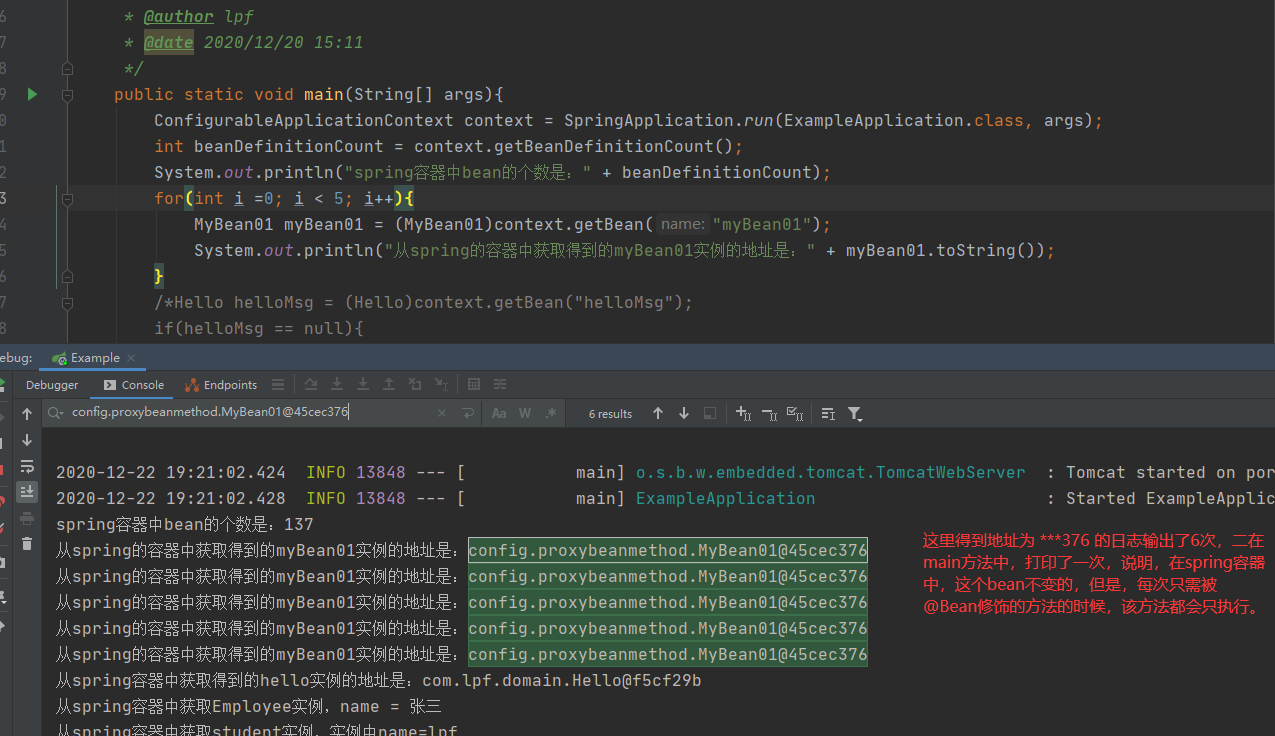
这里可以看到从myBean02 和myBean022 中获取得到的mybean01 的地址是一样的。虽然，他们在作为构造函数的参数传入的时候，调用了两次配置方法。

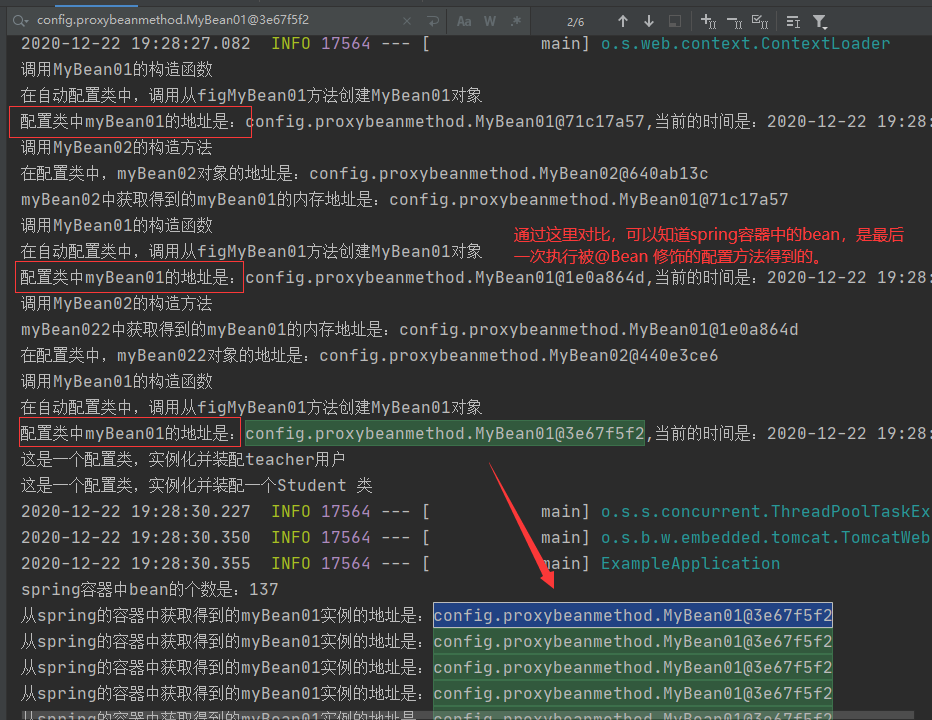


通过第二幅图和第三幅图，可以知道当@Configuration类的proxyBeanMethods 的取值为true的时候，被@Bean修饰的方法，只会执行一次，只会会将对应的bean放到spring容器当中。只会每次调用配置方法，都是从spring容器中去获取。



当@Configuration 注解的proxyBeanMethods的取值为false 的时候，每次，调用配置类，得到的bean都是从新创建的，每次得到的bean的内存地址都不一样。







## Springboot 中Spring mvc 添加http 缓存头提升响应性能

### Springboot 如何设置缓存头，来提供应用对静态资源的响应？

### Springboot中的模板引擎，如何使用缓存功能？？

### Spring.http.log-request-details 配置属性配置来查看请求的详细信息。

### 在你的application.properties中设置spring.devtools.addproperties 为false来禁用devtools 的默认属性。

### 类路径下文件改变的时候，springboot自动重启(前提，使用devtools)

这里，新增的方法，会不会生效(需要测试)

### SpringApplication.setRegisterShutdownHook(false) 使得重启不生效。

## Springboot中的触发文件可不可以有多个同名的，或者不同名的文件

因为springboot 中指定触发文件的时候，只指定了文件名，没有路径，那么不同路径下的同名文件名，哪一个会真正的起作用，还是说都会起作用？？

阅读： <https://www.jetbrains.com/help/idea/spring-boot.html#custom-configuration-files>

## 其他