目录

[spring注解驱动开发（规划） 1](#_Toc92662209)

[01- 组件注册@Configuration&@Bean给容器中注入组件（xml和注解配置bean） 2](#_Toc92662210)

[02-1使用xml的方式 2](#_Toc92662211)

[02-2使用注解的方式 2](#_Toc92662212)

[总结 2](#_Toc92662213)

[03、组件注册-@ComponentScan-自动扫描组件&指定扫描规则(xml中的包扫描和注解中的包扫描) 2](#_Toc92662214)

[03-1使用注解的方式查看容器中有那些组件 3](#_Toc92662215)

[03-2 包扫描的时候排除一些注解 3](#_Toc92662216)

[03-3 包扫描的时候只包含一些注解 4](#_Toc92662217)

[Xml中禁用扫描 4](#_Toc92662218)

[注解中禁用扫描 4](#_Toc92662219)

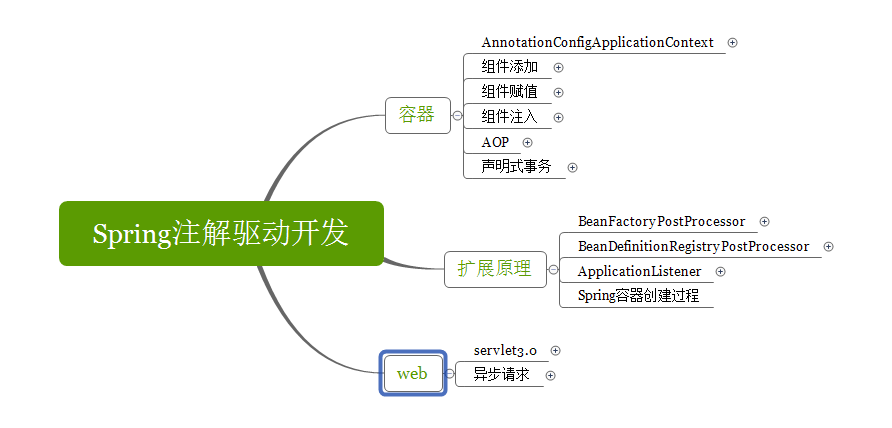
[总结 4](#_Toc92662220)

[04组件注册-自定义TypeFilter指定过滤规则 5](#_Toc92662221)

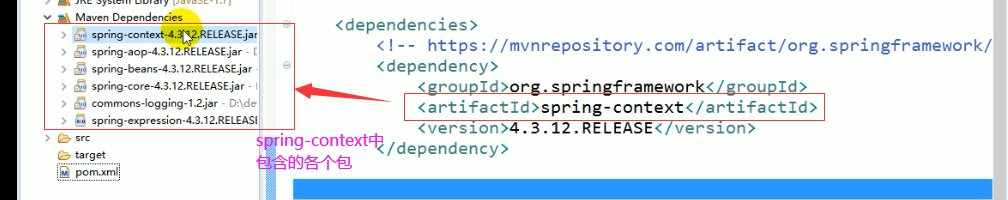
[自定义过滤器 5](#_Toc92662222)

[自定义过滤器的使用 5](#_Toc92662223)

# spring注解驱动开发（规划）



# 组件注册@Configuration&@Bean给容器中注入组件（xml和注解配置bean）



## 02-1使用xml的方式

com.atguigu.lianxi.Chapter2#test\_1

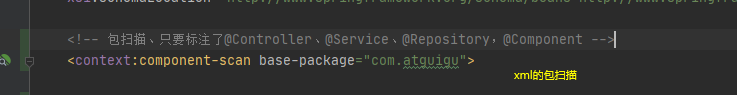
## 02-2使用注解的方式

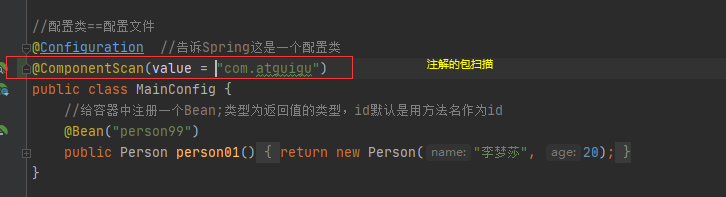
com.atguigu.lianxi.Chapter2#test\_2

## 总结

Bean的命名: @Bean注解中的value优先级高于方法名

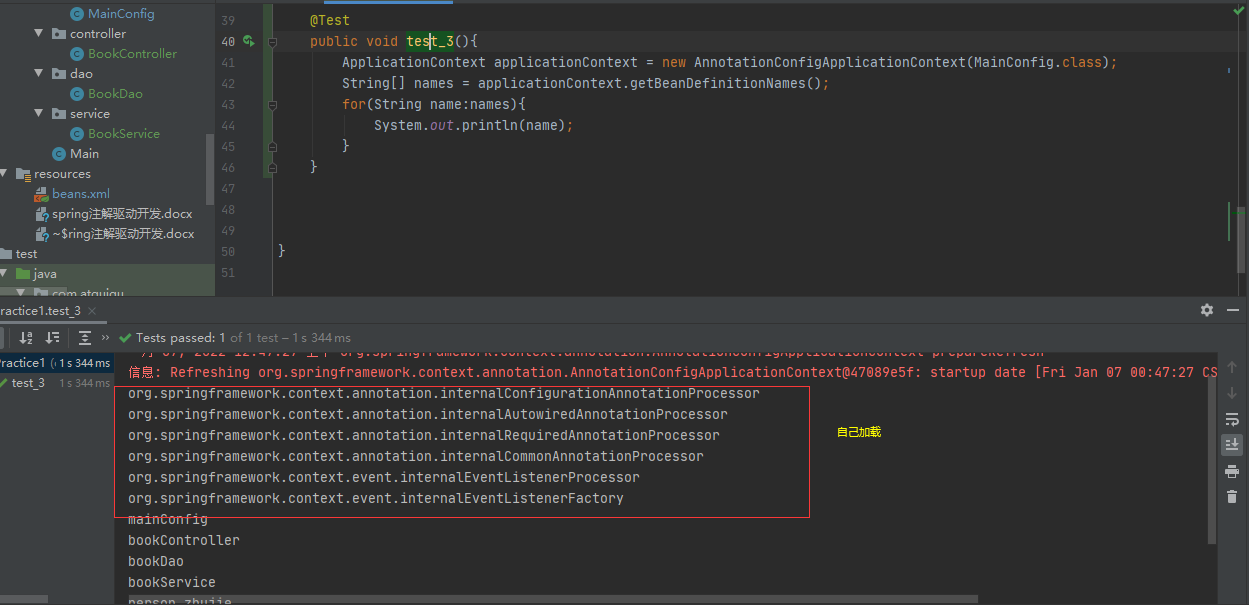
# 03、组件注册-@ComponentScan-自动扫描组件&指定扫描规则(xml中的包扫描和注解中的包扫描)



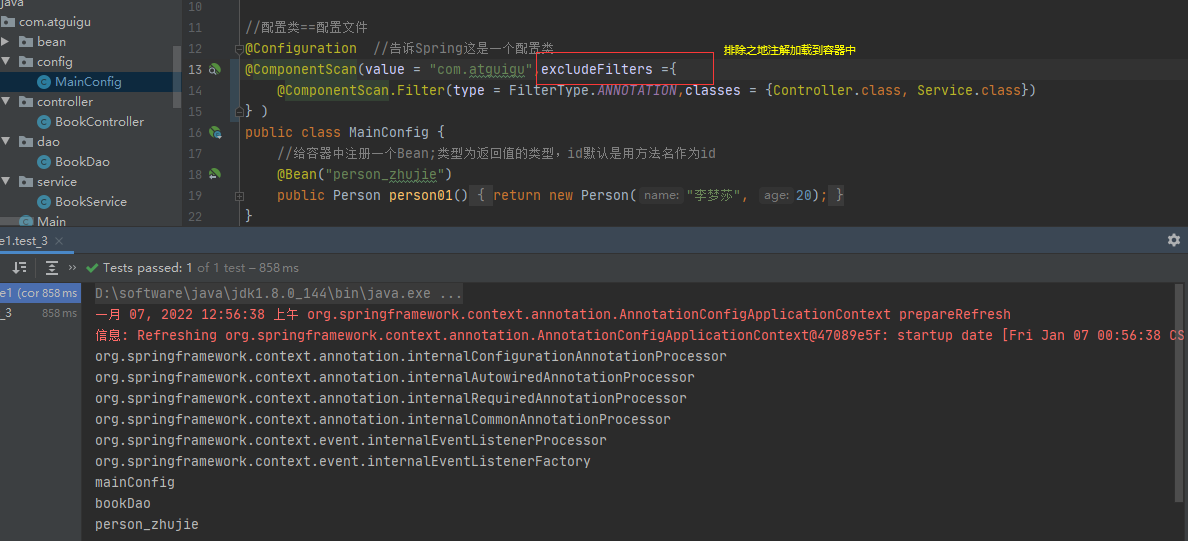


## 03-1使用注解的方式查看容器中有那些组件

com.atguigu.lianxi.Practice1#test\_3

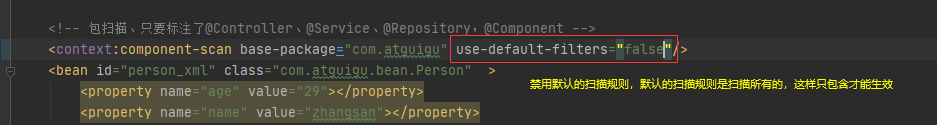


## 03-2 包扫描的时候排除一些注解

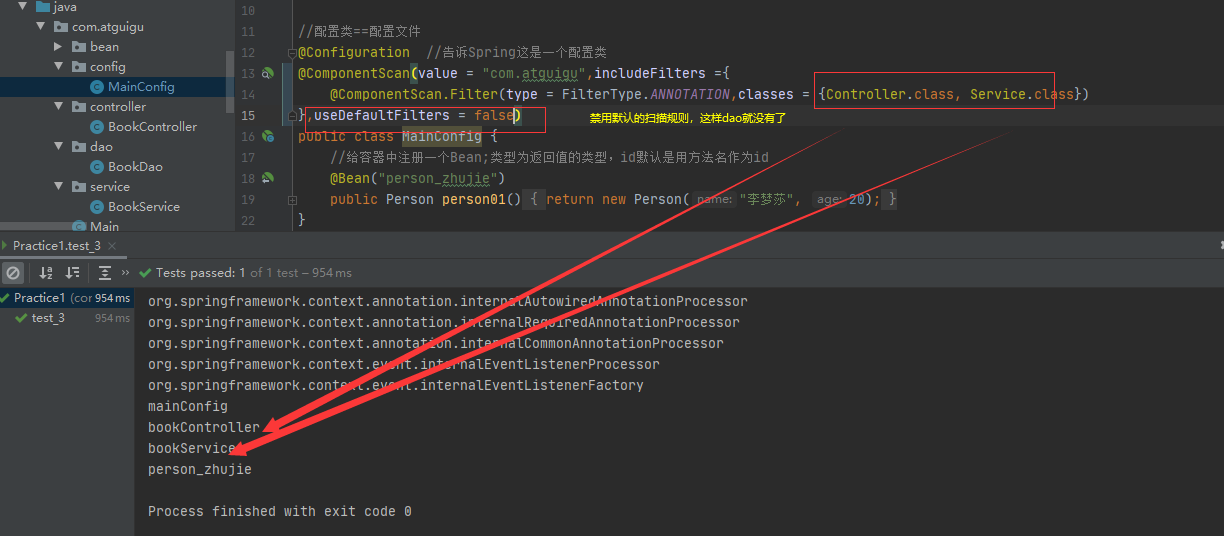


## 03-3 包扫描的时候只包含一些注解

### Xml中禁用扫描



### 注解中禁用扫描



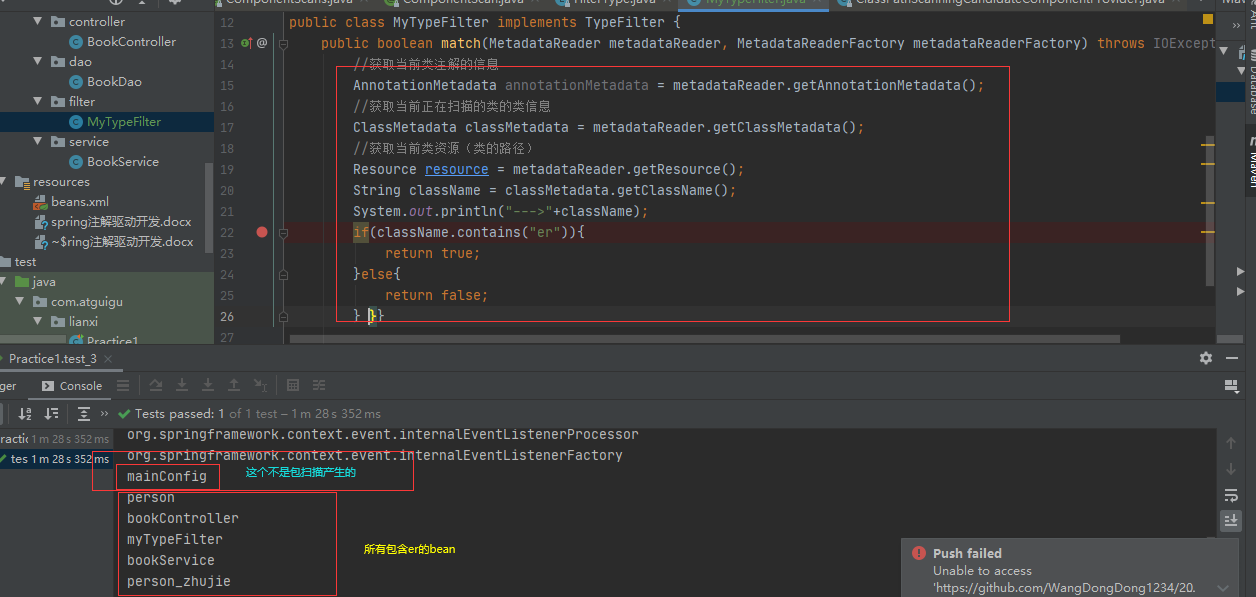
## 总结

//excludeFilters = Filter[] ：指定扫描的时候按照什么规则排除那些组件

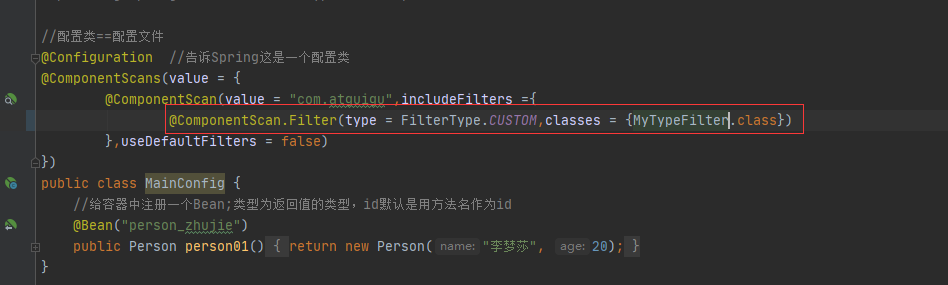
//includeFilters = Filter[] ：指定扫描的时候只需要包含哪些组件（必须禁用默认的扫描规则）

# 04组件注册-自定义TypeFilter指定过滤规则

## 自定义过滤器

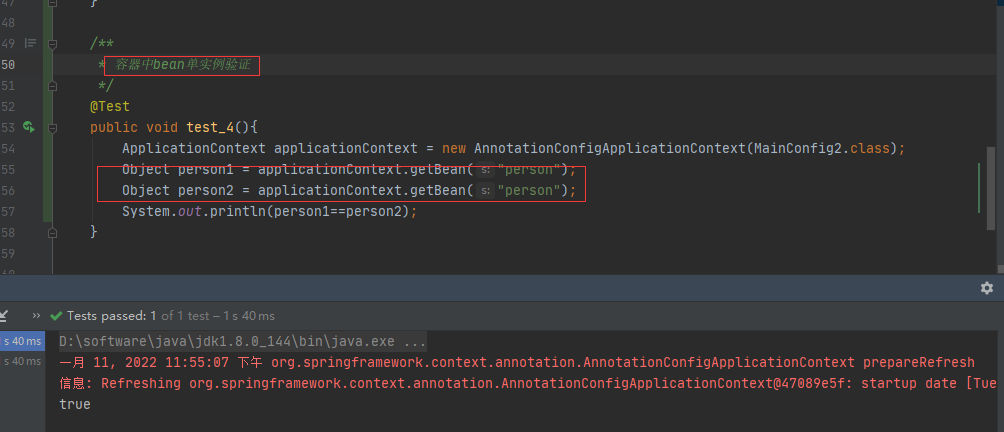


## 自定义过滤器的使用



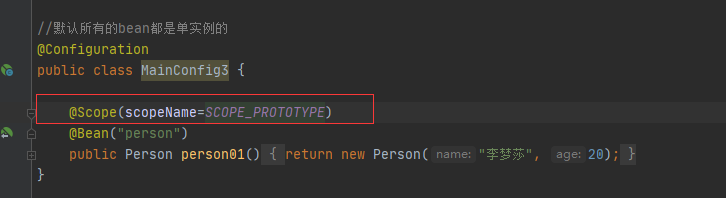
# 05组件注册@Scope设置组件作用域

## 05-1单实例验证（com.atguigu.lianxi.Practice1#test\_4）



@Scope等价于xml中的scope=””属性

## 05-2多实例验证（com.atguigu.lianxi.Practice1#test\_5）



## 

## 总结

prototype：多实例的：ioc容器启动并不会去调用方法创建对象放在容器中。每次获取的时候才会调用方法创建对象；

singleton：单实例的（默认值）：ioc容器启动会调用方法创建对象放到ioc容器中。以后每次获取就是直接从容器（map.get()）中拿，

request：同一次请求创建一个实例

session：同一个session创建一个实例

## 使用场景

1.对于有实例变量的类，要设置成prototype；没有实例变量的类，就用默认的singleton   
2.Action一般我们都会设置成prototype，而Service只用singleton就可以。

# 06、组件注册-@Lazy-bean懒加载(com.atguigu.lianxi.Practice1#test\_6)

懒加载（针对的是单实例bean）：

单实例bean：默认在容器启动的时候创建对象；

懒加载：容器启动不创建对象。第一次使用(获取)Bean创建对象，并初始化；

