# 配置文件值注入

## @ConfigurationProperties

### 功能

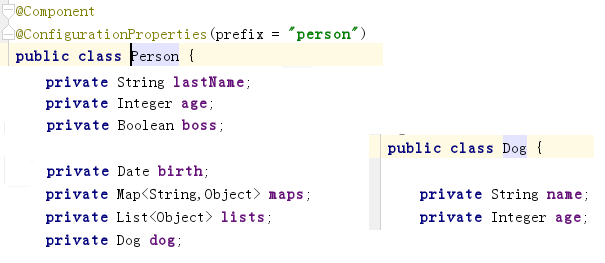
*将配置文件中配置的每一个属性的值，映射到这个组件中；****@ConfigurationProperties：告诉SpringBoot将本类中的所有属性和配置文件中相关的配置进行绑定；****prefix = "person"：配置文件中哪个下面的所有属性进行一一映射*

*只有这个组件是容器中的组件（*@Component*），才能容器提供的@ConfigurationProperties功能；****@ConfigurationProperties(prefix*** *= "person")默认从全局配置文件中获取值；*

### 实体类准备

看文件com.aiguigu.springboot.bean.PersonConfigurationProperties

注意这里不能是personProperties，必须是person-properties或者person\_properties



*将配置文件中配置的每一个属性的值，映射到这个组件中；****@ConfigurationProperties：告诉SpringBoot将本类中的所有属性和配置文件中相关的配置进行绑定；****prefix = "person"：配置文件中哪个下面的所有属性进行一一映射*

*只有这个组件是容器中的组件（*@Component*），才能容器提供的@ConfigurationProperties功能；****@ConfigurationProperties(prefix*** *= "person")默认从全局配置文件中获取值；*

### \*配置提示

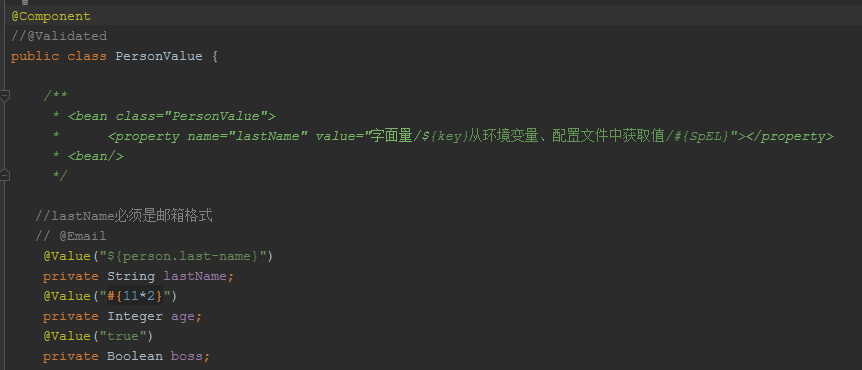
以导入配置文件处理器，以后编写配置就有提示了

*<!--导入配置文件处理器，配置文件进行绑定就会有提示-->*<**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-configuration-processor</**artifactId**>  
 <**optional**>true</**optional**>  
</**dependency**>

## @Value

### 实体类准备

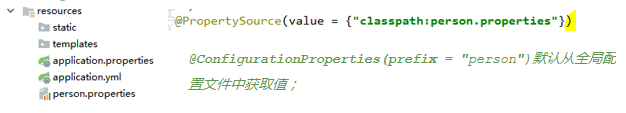
看文件com.aiguigu.springboot.bean.PersonValue



## @PropertySource（导入配置*properties*）

***@ConfigurationProperties(prefix*** *= "person")默认从全局配置文件中获取值；*

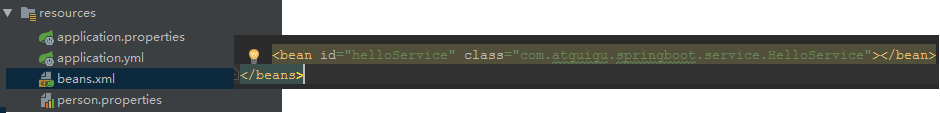
为了结构清晰，将配置文件拆分，这个注解的目的就是读取除了主配置文件的其他配置文件



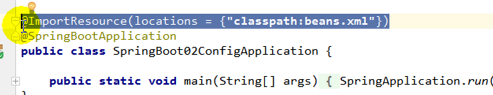
## @ImportResource（导入配置xml）



配置文件xml



导入



测试：看文件com.atguigu.springboot.SpringBoot02ConfigApplicationTests的方法testHelloService

返回结果true

## @Bean(注解方式)--取代@importResouce

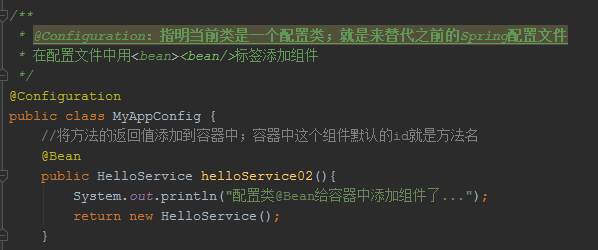
不来编写Spring的配置文件

SpringBoot推荐给容器中添加组件的方式；推荐使用全注解的方式

1. 配置类@Configuration------>Spring配置文件

***@Configuration：指明当前类是一个配置类；就是来替代之前的Spring配置文件***

2、使用@Bean给容器中添加组件



测试：看文件com.atguigu.springboot.SpringBoot02ConfigApplicationTests的方法testHelloService

返回结果

# Springboot测试

## 测试注解

看文件com.aiguigu.springboot.SpringBoot02ConfigApplicationTests



## 测试

看文件：com.aiguigu.springboot.SpringBoot02ConfigApplicationTests

@ConfigurationProperties 看文件contextLoads2()



@Value:不支持复杂类型 看文件contextLoads3()



# @Value和@ConfigurationProperties

## @Value和@ConfigurationProperties获取值比较



配置文件yml还是properties他们都能获取到值；

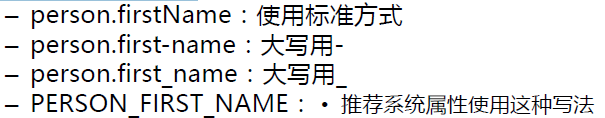
如果说，我们只是在某个业务逻辑中需要获取一下配置文件中的某项值，使用@Value；

如果说，我们专门编写了一个javaBean来和配置文件进行映射，我们就直接使用@ConfigurationProperties；

## @ConfigurationProperties

### 支持松散语法绑定

属性名匹配规则



1.值里last-name和lastName 同义

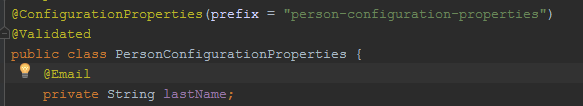


2.实体类不能是personProperties，必须是person-properties或者person\_properties



### 支持JSR303数据校验-- Validation

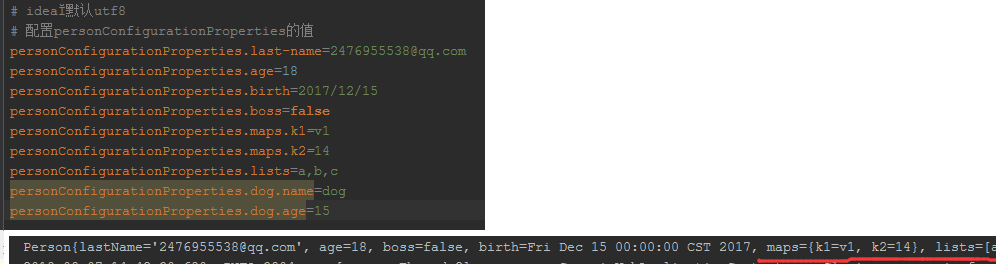
（@ConfigurationProperties Validation）



### 支持复杂类型封装

可以解析map，list

而@value只能



## @Value

### 支持spel



### \*全局不行，局部可以

我也不晓得为啥子

