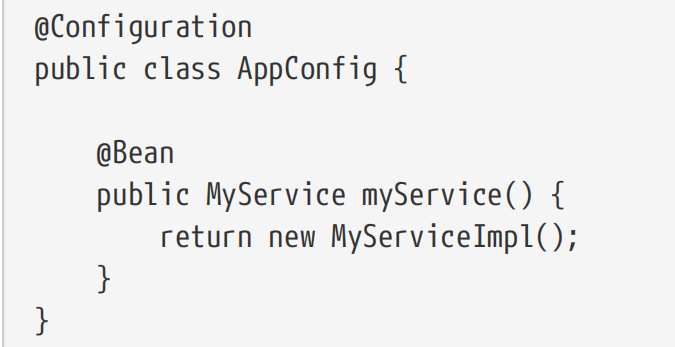
# IOC容器

Package：Bean & Context

## Java-based Container Configuration

### @Bean and @Configuration



效果等价于

<beans>

  <bean id="myService" class="com.acme.services.MyServiceImpl"/>

</beans>

Annotating a class with @Configuration

* indicates that its primary purpose is as a source of bean definitions.
* let inter-bean dependencies be defined by calling other @Bean methods in the same class.

### AnnotationConfigApplicationContext

capable of accepting not only @Configuration classes as input but also plain @Component classes and classes annotated with JSR-330 metadata.

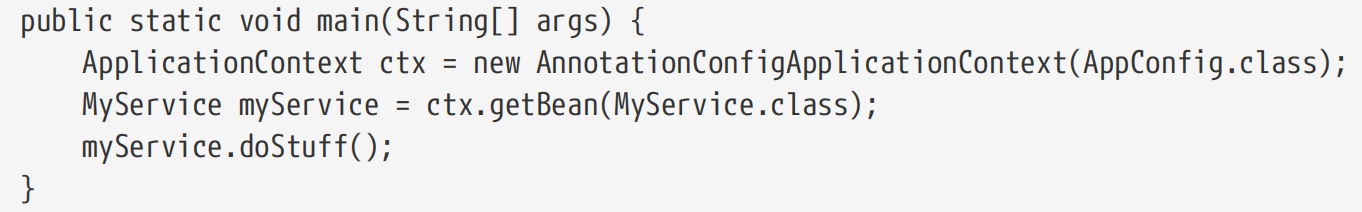
When @Configuration classes are provided as input, the @Configuration class itself is registered as a bean definition and all declared @Bean methods within the class are also registered as bean

definitions.

XML格式：ClassPathXmlApplicationContext

@Configuration classes：AnnotationConfigApplicationContext，不再需要使用XML

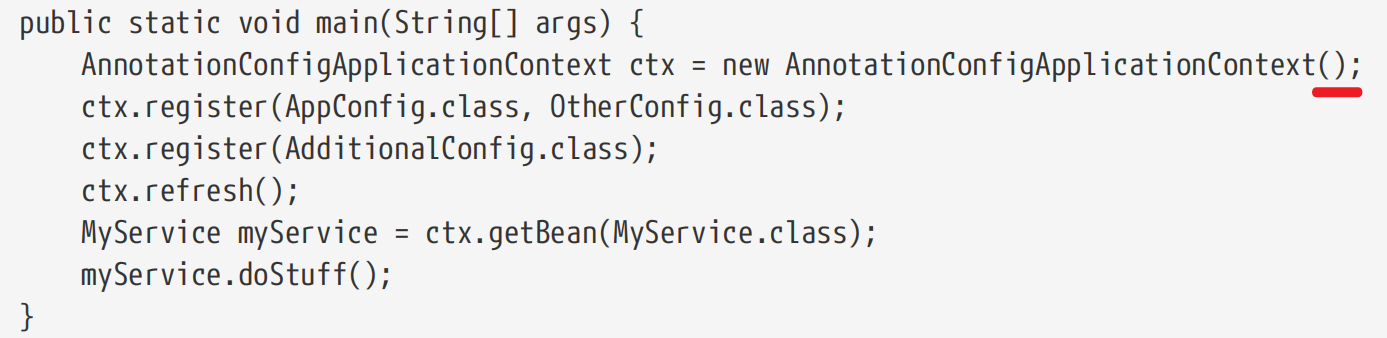
场景1：传入configuration类



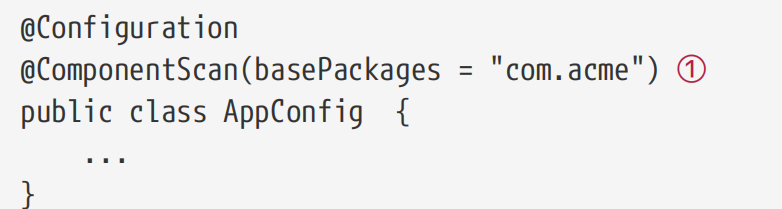
场景2：@Component和JSR330注解类



场景3：空构造函数+register()+refresh()



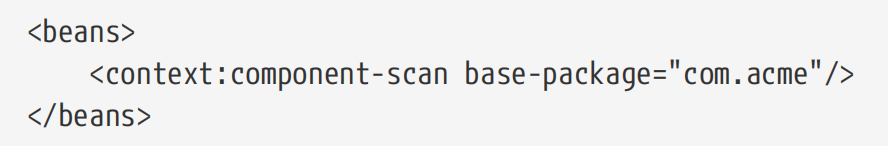
@ComponentScan(basePackages = "com.acme")



其他实现方案： refresh()用于处理注册bean定义



效果XML定义



### AnnotationConfigWebApplicationContext

# Bean校验校验

## 规范

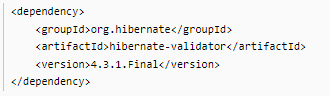
JSR-303 1.0

JSR 349 1.1

JSR 380 2.0

## Hibernate Validator

实现Bean Validation规范



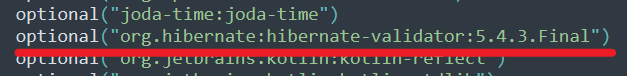


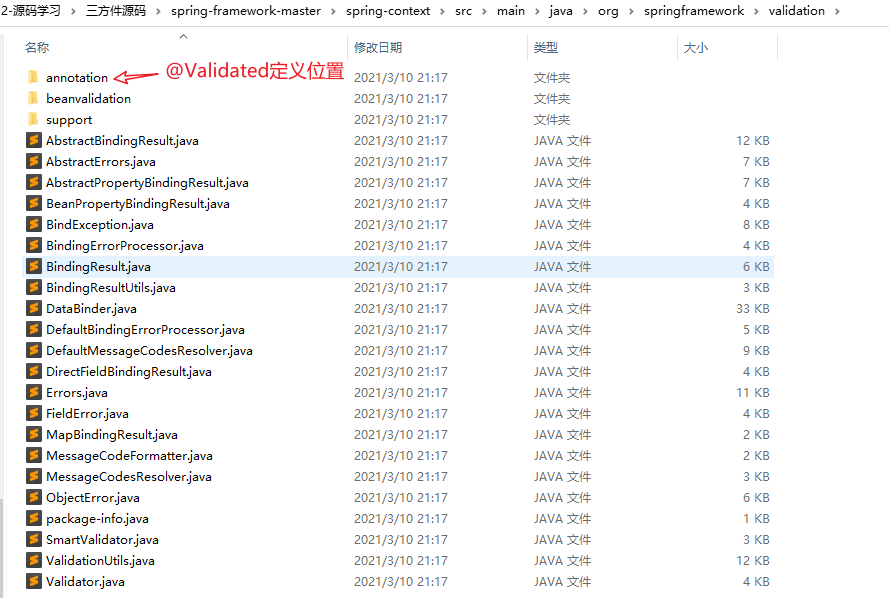
使用方式



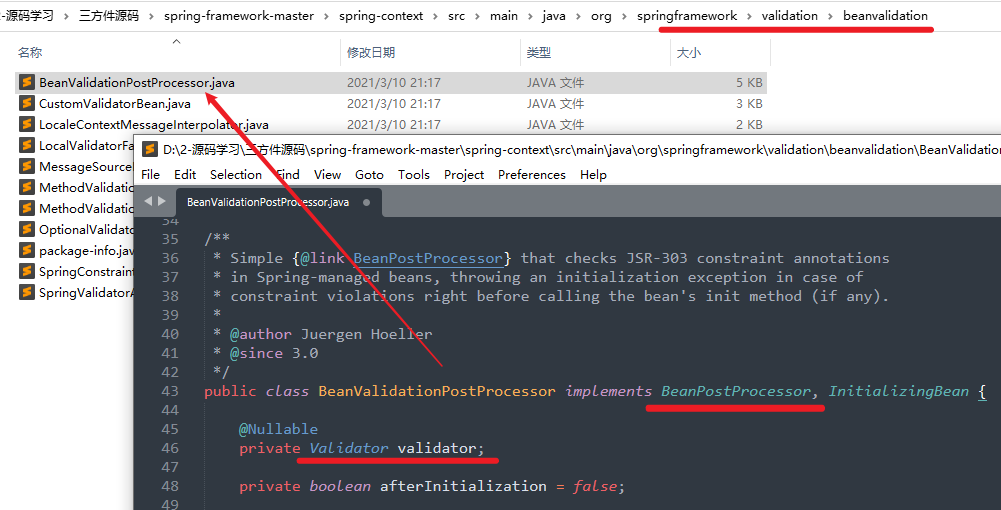
## Spring-context定义的validator实现

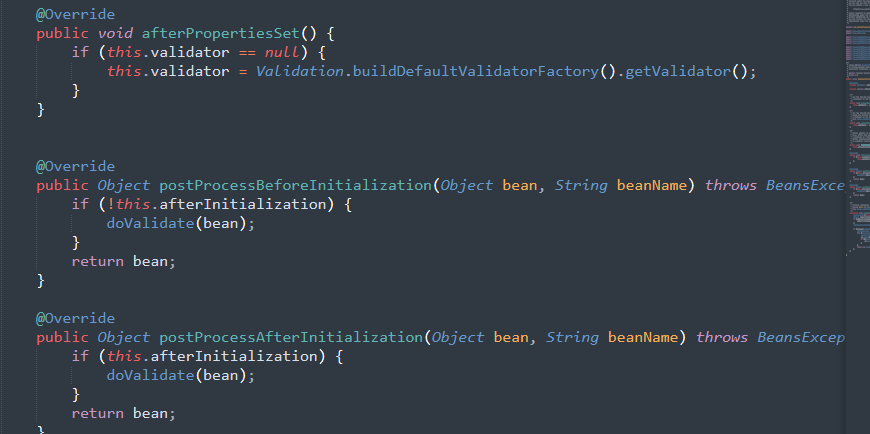
build.gradle





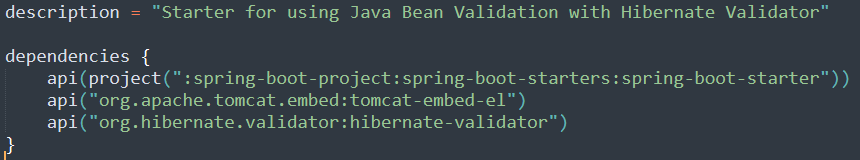
**BeanValidationPostProcessor: 在bean初始化时通过Validator进行校验**





## spring-boot-starter-validation

build.gradle定义



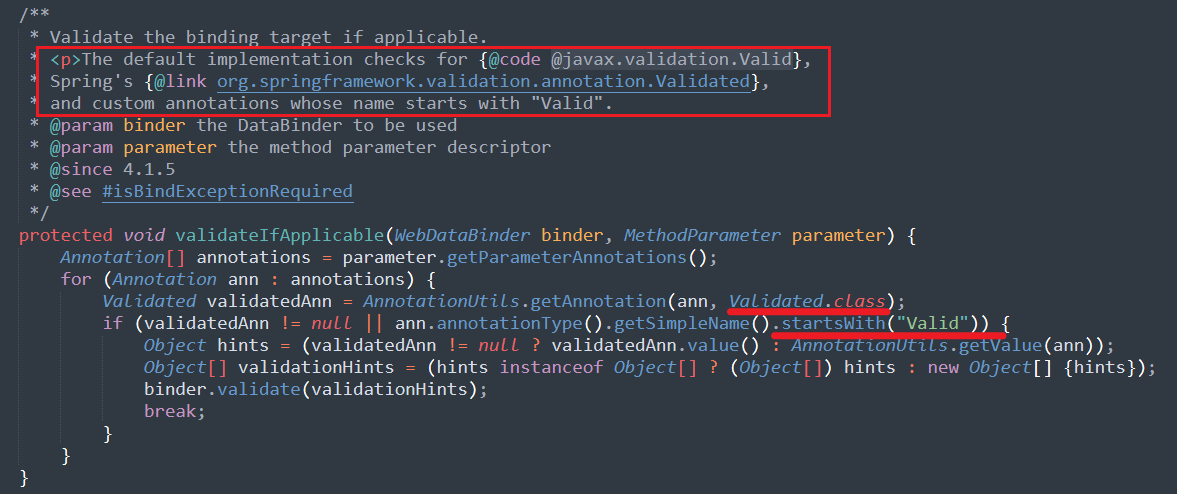
样例

1. Bean定义字段校验规则：
2. Controller引入@Valid或@Validator触发校验



@Valid和@Validate都可触发校验的原理

org.springframework.web.servlet.mvc.method.annotation.AbstractMessageConverterMethodArgumentResolver



## 开发过程

　　1、Bean定义字段校验规则：

　　2、Controller引入@Valid（来自）或@Validated（来自org.springframework.validation.annotation）触发校验

　　样例如下：



**国际化校验提示**

