链滴

SpringBoot 开发技巧 - 启动时配置校验

作者：jianzh5

原文链接：https://ld246.com/article/1598844125326

来源网站：链滴

概述

在项目开发过程中，某个功能需要依赖在配置文件中配置的参数。这时候就可能出现下面这种现象问：

有时候经常出现项目启动了，等到使用某个功能组件的时候出现异常，提示参数未配置或者bean注失败。

有没有一种方法在项目启动时就对参数进行校验而不是在实际使用的时候再抛出提示呢?

答案就是使用Spring提供的Java Validation功能，简单实用。

增加启动校验

只需要在我们创建的配置Properties类增加Validation相关配置即可

```
@Validated
@Data
@ConfigurationProperties(prefix = "app")
@Component
public class AppConfigProperties {
    @NotEmpty(message = "配置文件配置必须要配置[app.id]属性")
    private String id;
}
```

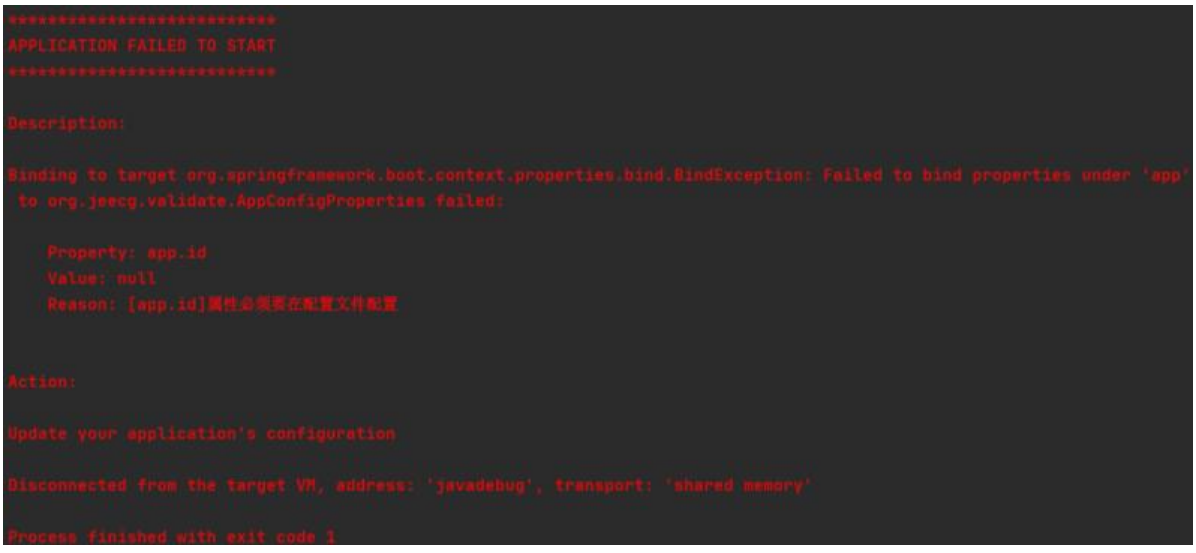
上面的配置就会校验我们在 application.yml中有没有配置 app.id参数。如果在配置文件中没有该配，项目启动就会失败，并抛出校验异常。

在使用配置文件校验时，必须使用@configurationproperties注解，@value不支持该注解。

在需要使用app.id的时候注入配置类即可：

```
@Autowired
private AppConfigProperties appConfigProperties;
```

这样就可以实现我们想要的效果，如下图：



校验类型

校验规则	规则说明
@Null	限制只能为null
@NotNull	限制必须不为null
@AssertFalse	限制必须为false
@AssertTrue	限制必须为true
@DecimalMax(value)	限制必须为一个不大于指值的数字
@DecimalMin(value)	限制必须为一个不小于指值的数字
@Digits(integer,fraction)	限制必须为一个小数且整数部分的位数不能超过integer，小数部分的位数不能超过fraction
@Future	限制必须是一个将来的日期
@Max(value)	限制必须为一个不大于指定值的数字
@Min(value)	限制必须为一个不小于指定值的数字
@Past	验证注解的元素值（日期类型）比当前时早
@Pattern(value)	限制必须符合指定的正则表达式
@Size(max,min)	限制字符长度必须在min到ma之间
@NotEmpty	验证注解的元素值不为null且不为（字符串长度不为0、集合大小不为0）
@NotBlank	验证注解的元素值不为空（不为null去除首位空格后长度为0），不同于@NotEmpty，@NotBlank只应用于字符串且在比较时会去除字串的空格
@Email	验证注解的元素值是Email，也可以通过则表达式和flag指定自定义的email格式

Validation 支持如下几种校验，可以满足基本的业务逻辑，当然如果还是满足不了你的业务逻辑，可选择定制校验规则。

定制校验逻辑

1. 定义校验逻辑规则，实现 org.springframework.validation.Validator

```
public class ConfigPropertiesValidator implements Validator {
    @Override
    public boolean supports(Class<?> aClass) {
        return AppConfigProperties.class.isAssignableFrom(aClass);
    }

    @Override
    public void validate(Object o, Errors errors) {
        AppConfigProperties config = (AppConfigProperties) o;
```

```
        if(StringUtils.isEmpty(config.getId())){
            errors.rejectValue("id", "app.id.empty", "[app.id] 属性必须要在配置文件配置");
        }else if (config.getId().length() < 5) {
            errors.rejectValue("id", "app.id.short", "[app.id] 属性的长度必须不能小于5");
        }
    }
}
```

2. 使用自定义校验规则就不需要在使用原生的@NotEmpty了，将其删除

```
@Validated
@Data
@ConfigurationProperties(prefix = "app")
@Component
public class AppConfigProperties {
//    @NotEmpty(message = "配置文件配置必须要配置[app.id]属性")
    private String id;
}
```

3. 注入自定义校验规则

```
@Bean
public  ConfigPropertiesValidator configurationPropertiesValidator(){
  return new ConfigPropertiesValidator();
}
```

注意：这里bean的方法名必须要 configurationPropertiesValidator，否则启动的时候不会执行校验

4. 修改app.id配置，观察启动情况

错误信息即为我们自定义校验的结果。

小结

通过配置Spring Boot启动校验功能，可以快速的识别参数配置的错误，避免在使用组件的时候才发问题，可以减少排查问题的工作量，并且在我们封装自定义的starter时可以有更好的体验。