SpringBoot @Constraint自定义注解

这可真叫人头秃 2020-03-18 16:07:39 937 收藏 8

版权

SpringBoot有四个注解用来注解注解的@Retention``````@Target``````@Document``````@Inherited

注解保留位置的：@Retention

//注解仅存在于源码中，在class字节码文件中不包含

@Retention(RetentionPolicy.SOURCE)

// 默认的保留策略，注解会在class字节码文件中存在，但运行时无法获得

@Retention(RetentionPolicy.CLASS)

// 注解会在class字节码文件中存在，在运行时可以通过反射获取到

@Retention(RetentionPolicy.RUNTIME)

注解的作用目标：@Target

//接口、类、枚举、注解

@Target(ElementType.TYPE)

//字段、枚举的常量

@Target(ElementType.FIELD)

//方法

@Target(ElementType.METHOD)

//方法参数

@Target(ElementType.PARAMETER)

//构造函数

@Target(ElementType.CONSTRUCTOR)

//局部变量

@Target(ElementType.LOCAL\_VARIABLE)

//注解

@Target(ElementType.ANNOTATION\_TYPE)

//包

@Target(ElementType.PACKAGE)

将注解包含在javadoc中@Document

注解可以被继承@Inherited

自定义注解@Constraint

如果想对接口参数进行校验，必须使用@valid才能使注解生效

@GetMapping("/check")

public String test02(@Valid @RequestBody Parm s){

return s.getNameS();

}

下面就是如何具体完成一个自定义注解，首先是定义一个@interface

@Constraint(validatedBy= {CheckNameValidator.class})

@Retention(RetentionPolicy.RUNTIME)

@Target(ElementType.FIELD)

public @interface CheckName {

String message();

Class<?>[] groups() default { };

Class<? extends Payload>[] payload() default { };

}

其中，CheckName为注解名，validatedBy= {CheckNameValidator.class}的CheckNameValidator是真正进行校验方法的类，message可自定义个数和属性，作为注解传参的参数

注意，这两个必须有，否则报错：

Class<?>[] groups() default { };

Class<? extends Payload>[] payload() default { };

真正校验的类：

public class CheckNameValidator implements ConstraintValidator<CheckName, String> {

String message;

@Override

public void initialize(CheckName constraintAnnotation) {

this.message = constraintAnnotation.message();

}

@Override

public boolean isValid(String s, ConstraintValidatorContext constraintValidatorContext) {

if (s.equals("a")) {

System.out.println(message);

return true;

}

return false;

}

}

需要继承ConstraintValidator<注解名，校验的Object>

有两个方法initialize()和isValid()

方法initialize()不需要一定有，使用可用来对注解定义的参数进行初始化给isValid()方法进行使用

方法isValid()必须实现，是校验逻辑所在的位置

最后是注解的位置：

public class Parm {

@CheckName(message = "message")

private String NameS;

public String getNameS() {

return NameS;

}

public void setNameS(String name) {

NameS = name;

}

}

注解括号里的是注解中定义的参数，可自定义

————————————————

版权声明：本文为CSDN博主「这可真叫人头秃」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq\_38425719/article/details/104946921