**使用注解**

一、准备校验时使用的JAR

validation-api-1.0.0.GA.jar：JDK的接口；

hibernate-validator-4.2.0.Final.jar是对上述接口的实现；

log4j、slf4j、slf4j-log4j

二、编写需要校验的bean

@NotNull(message="名字不能为空")private String userName;

@Max(value=120,message="年龄最大不能查过120")private int age;

@Email(message="邮箱格式错误")private String email;

三、校验方法

[IMG_256](http://www.cnblogs.com/yangzhilong/p/javascript:void(0);)

@RequestMapping("/login")

public String testValid(@Valid User user, BindingResult result){

if (result.hasErrors()){

List<ObjectError> errorList = result.getAllErrors();

for(ObjectError error : errorList){

System.out.println(error.getDefaultMessage());

}

}

return "test";

}

[IMG_257](http://www.cnblogs.com/yangzhilong/p/javascript:void(0);)

备注：这里一个@Valid的参数后必须紧挨着一个BindingResult 参数，否则spring会在校验不通过时直接抛出异常

前台可以使用spring的标签库也可以自己自定义处理

spring标签库的用法：

[IMG_258](http://www.cnblogs.com/yangzhilong/p/javascript:void(0);)

<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>

<html> <head> <title>Reservation Form</title> <style>

.error {

color: #ff0000;

font-weight: bold; } </style> </head>

<body>

<form:form method="post" modelAttribute="vm">

<form:errors path="\*" cssClass="error" />

<table>

<tr>

<td>Name</td>

<td><form:input path="userName" />

</td>

<td><form:errors path="userName" cssClass="error" />

</td>

</tr>

<tr>

<td>email</td>

<td><form:input path="email" />

</td>

<td><form:errors path="email" cssClass="error" />

</td>

</tr>

<tr>

<td colspan="3"><input type="submit" />

</td>

</tr>

</table>

</form:form> </body> </html>

[IMG_259](http://www.cnblogs.com/yangzhilong/p/javascript:void(0);)

四、开启spring的Valid功能

<mvc:annotation-driven />

五、JSR303定义的校验类型

[IMG_260](http://www.cnblogs.com/yangzhilong/p/javascript:void(0);)

空检查

@Null       验证对象是否为null

@NotNull    验证对象是否不为null, 无法查检长度为0的字符串

@NotBlank 检查约束字符串是不是Null还有被Trim的长度是否大于0,只对字符串,且会去掉前后空格.

@NotEmpty 检查约束元素是否为NULL或者是EMPTY.

Booelan检查

@AssertTrue     验证 Boolean 对象是否为 true

@AssertFalse    验证 Boolean 对象是否为 false

长度检查

@Size(min=, max=) 验证对象（Array,Collection,Map,String）长度是否在给定的范围之内

@Length(min=, max=) Validates that the annotated string is between min and max included.

日期检查

@Past           验证 Date 和 Calendar 对象是否在当前时间之前

@Future     验证 Date 和 Calendar 对象是否在当前时间之后

@Pattern    验证 String 对象是否符合正则表达式的规则

数值检查，建议使用在Stirng,Integer类型，不建议使用在int类型上，因为表单值为“”时无法转换为int，但可以转换为Stirng为"",Integer为null

@Min            验证 Number 和 String 对象是否大等于指定的值

@Max            验证 Number 和 String 对象是否小等于指定的值

@DecimalMax 被标注的值必须不大于约束中指定的最大值. 这个约束的参数是一个通过BigDecimal定义的最大值的字符串表示.小数存在精度

@DecimalMin 被标注的值必须不小于约束中指定的最小值. 这个约束的参数是一个通过BigDecimal定义的最小值的字符串表示.小数存在精度

@Digits     验证 Number 和 String 的构成是否合法

@Digits(integer=,fraction=) 验证字符串是否是符合指定格式的数字，interger指定整数精度，fraction指定小数精度。

@Range(min=, max=) Checks whether the annotated value lies between (inclusive) the specified minimum and maximum.

@Range(min=10000,max=50000,message="range.bean.wage")  
private BigDecimal wage;

@Valid 递归的对关联对象进行校验, 如果关联对象是个集合或者数组,那么对其中的元素进行递归校验,如果是一个map,则对其中的值部分进行校验.(是否进行递归验证)

@CreditCardNumber信用卡验证

@Email  验证是否是邮件地址，如果为null,不进行验证，算通过验证。

@ScriptAssert(lang= ,script=, alias=)

@URL(protocol=,host=, port=,regexp=, flags=)