

对非String类型进行校验时的出错处理 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/144/2021\\_2022\\_\\_E5\\_AF\\_B9\\_E9\\_9D\\_9EStri\\_c104\\_144772.htm](https://www.100test.com/kao_ti2020/144/2021_2022__E5_AF_B9_E9_9D_9EStri_c104_144772.htm) 关于Spring MVC对非String类型

进行校验时的出错处理 我在对User.age (Integer型),进行校验时,如果输入的是字符总是出错: Failed to convert property value of type [java.lang.String] to required type [java.lang.Integer] for property age. nested exception is

java.lang.NumberFormatException: For input string: "ff" 解决方法:

1.本例用的是commons-validator的开源框架 2.在Controller中加

```
protected void initBinder(HttpServletRequest request,
ServletRequestDataBinder binder) {
```

```
binder.registerCustomEditor(Integer.class, null, new
CustomNumberEditor(Integer.class, null, true)).
```

```
binder.registerCustomEditor(Long.class, null, new
CustomNumberEditor(Long.class, null, true)).
```

```
binder.registerCustomEditor(byte[].class, new
ByteArrayMultipartFileEditor()). SimpleDateFormat dateFormat =
new SimpleDateFormat(getText("date.format",
request.getLocale())). dateFormat.setLenient(false).
```

```
binder.registerCustomEditor(Date.class, null, new
CustomDateEditor(dateFormat, true)). } 3.在资源文
```

件ApplicationResources\_zh\_CN.properties 中加:

```
user.age=\u5e74\u9f84
```

```
typeMismatch.java.lang.Integer={0}\u5fc5\u987b\u4e3a\u4e00\u65
70\u503c\u3002 4.一些朋友说在用非对象类型如int时也总是出
```

错,但我在测试时发现不管是用int 还是用float,如果不输或输入字符,系统均会自动赋值 int为0,float为0.0 5.以下是一个不错的入门文章,转发于 " jamyy2000的专栏 " :关于Spring 中Commons Validator的使用说明 spring的validate验证需要两个bean ( ValidatorFactory、 BeanValidator ) 及两xml文件 ( validator-rules.xml、 validation.xml ) 的支持 , ValidatorFactory 用于制造BeanValidator , 后者则是在程序中执行校验的控制bean。 Validatorrules.xml中定义了各种验证的规则 , 如字段不为空 , 字段输入值必须为整数等等。 Validation.xml中则定义了那些bean及bean中的哪些属性字段需要验证 , 使用哪些验证规则。 validate 分为两级验证 , 客户端验证 ( javascript ) 和服务器端验证。 以下是使用validate的步骤 : 1. 在配置文件中声明ValidatorFactory 和 BeanValidator : -----

```
<!-- /WEB-INF/validator-rules.xml /WEB-INF/validation.xml -->
<bean class="org.springframework.validation.beanvalidation.DefaultValidatorFactory"
       validationConfigLocations="classpath:/WEB-INF/validator-rules.xml,classpath:/WEB-INF/validation.xml" />
```

声明validatorFactory这里我们使用spring 的DefaultValidatorFactory 定义其validationConfigLocations属性 , 将validator-rules.xml 和validation.xml传入 声明beanValidator这里我们使用spring 的DefaultBeanValidator 在其属性中指明要使用的validatorFactory。 这里我们使用刚刚定义的validatorFactory 2. 在需要进行验证的controller ( 即要使用form表单或者command的controller ) 中声明validate。 -----

```
<!-- ----- -->
<commandName="commandName"
       validationConfigLocations="classpath:/WEB-INF/validator-rules.xml,classpath:/WEB-INF/validation.xml"
       commandClass="commandClass" />
```

commandName用于指明需要验证的command的名字 , 这个名字必须和 validation.xml中所写的名字保持一致。 commandClass用于指定这个command的类型

。其必须与你jsp提交的form最后形成的command类型相一致。  
。 声明该controller使用validator，这里将我们刚刚定义的beanValidator传入进去。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)