## 一. spring自带的服务器校验(原生校验)

1. 创建类实现Validator接口，加上@Component注解
2. 实现方法Validator接口的两个方法

* validate方法进行验证错误信息加入Errors对象中， key=属性名 value=**国际化资源文件**中的key

public boolean supports(Class<?> aClass){  
 return aClass.isAssignableFrom(需要验证的类.class);  
}   
public void validate(Object o,Errors error){  
 error.rejectValue("属性名","错误信息(可以是国际化资源)")  
}

1. 将实现类注入到controller类中，加上@Autowired

* @Autowired  
  private 实现类 xxx;

1. 在controller方法中声明Errors对象,**且Errors对象必须紧跟model attribute（实体类）对象后面**，在Controller方法中判断errors对象里面是否有错误信息 errors.hasErrors()。

* 有：准备数据，返回注册页面   
  否：进行业务方法调用

@RequestMapping(value = "saveUser.action", method = RequestMethod.POST)  
public String saveUser(@ModelAttribute() User user, Errors errors, Model model) {  
 userValidater.validate(user, errors);  
 //判断是否存在错误  
 if (errors.hasErrors()) {  
 this.setData(model);  
 model.addAttribute("user",user);  
 return "register";  
 }  
 return "register";  
}

Errors对象与当前user对象进行了绑定,如果此时在方法中添加新的user对象，那么前端无法获取errors的错误信息。

1. 在jsp中用<form:errors path="对象的属性名">

## 二. JSR303服务器校验

jsr303验证  
 a、加入jar包  
 b、springmvc.xml中注册校验器  
 c、修改controller，@Valid User user  
 d、Bean属性上使用注解设置验证规则

1. 配置文件 pom.xml

<dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-validator</artifactId>  
 <version>5.4.1.Final</version>  
</dependency>

springmvc.xml

<!--jsr303验证，声明验证器-->  
<bean id="localValidatorFactoryBean" class="org.springframework.validation.beanvalidation.LocalValidatorFactoryBean">  
 <property name="providerClass" value="org.hibernate.validator.HibernateValidator"/>  
 <!--注入国际化的bean(messageSource),国际化配置在之前的小结有-->  
 <property name="validationMessageSource" ref="messageSource"/>  
</bean>  
  
<!--解析注解-->  
<mvc:annotation-driven conversion-service="conversionService" validator="localValidatorFactoryBean"/>

1. 配置需要验证的类,在类的属性名上配置验证的方式

* @NotBlank(message="{name.error.notblank}") --- 使用国际化文件中的提示信息(需要加上一个括号)
* @NotBlank(message="提示") --- 直接写提示信息

1. 在Controller的方法中

* public void fun(@Valid Pojo pojo,Errors errors){}  
  (必须要在需要校验的类前加上Valid注解，并且后面必须要紧接着声明Errors对象)

1. 常用的校验API
   1. 空检查  
      @Null：验证对象是否为nul
   * @NotNull：验证对象是否不为null，**但可以为empty,没有Size的约束**，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后的长度是否大于0，只对字符串，且会去掉前后空格
   * @NotEmpty：**用在集合类上面**加了@NotEmpty的String类、Collection、Map、数组，是不能为null或者长度为0的(String Collection Map的isEmpty()方法)  
     检查约束元素是否为null或者是empty
     1. booelan检查  
        @AssertTrue：验证boolean属性是否为true
     + @AssertFalse：验证boolean属性是否为false
     1. 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=)：验证字符串长度是否在给定的范围之内
     1. 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
     1. 数值检查  
        @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