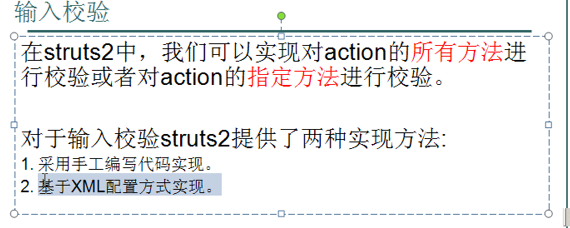
# 一。校验

# 1. 对action所有的方法进行校验



## 1.配置html <s:fielderror/>用于接收错误的信息

<%@ taglib uri=*"/struts-tags"* prefix=*"s"*%>

<s:fielderror/>

<form action=*"*${pageContext.request.contextPath}*/person/manage\_save.action"* method=*"post"*>

用户名:<input type=*"text"* name=*"username"*/>不能为空<br/>

手机号:<input type=*"text"* name=*"mobile"*/>不能为空,并且要符合手机号的格式1,3/5/8,后面是9个数字<br/>

<input type=*"submit"* value=*"提 交"*/></form>

## 2.配置action 继承ActionSupport{

**public** **class** PersonAction **extends** ActionSupport{

**private** String username;

**private** String mobile;

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getMobile() {

**return** mobile;

}

**public** **void** setMobile(String mobile) {

**this**.mobile = mobile;

}

**public** String update(){

ActionContext.*getContext*().put("message", "更新成功");

**return** "message";

}

**public** String save(){

ActionContext.*getContext*().put("message", "保存成功");

**return** "message";

}

@Override

**public** **void** validate() {//会对action中的所有方法校验

**if**(**this**.username==**null** || "".equals(**this**.username.trim())){

System.*out*.println(**this**.username.trim()+"trim处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

**this**.addFieldError("username", "用户名不能为空");

}

**if**(**this**.mobile==**null** || "".equals(**this**.mobile.trim())){

**this**.addFieldError("mobile", "手机号不能为空");

}**else**{

//正则表达式^1

**if**(!Pattern.*compile*("^1[358]\\d{9}$").matcher(**this**.mobile).matches()){

**this**.addFieldError("mobile", "手机号格式不正确");

}

}

}

}

## 3.配置Struts <!-- 校验错误会到input的界面 -->

<package name=*"person"* namespace=*"/person"* extends=*"struts-default"*>

<action name=*"manage\_\*"* class=*"cn.itcast.action.PersonAction"* method=*"{1}"*>

<!-- 校验错误会到input的界面 -->

<result name=*"input"*>/index.jsp</result>

<result name=*"message"*>/WEB-INF/page/message.jsp</result>

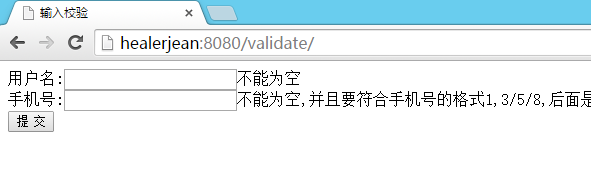
</action>

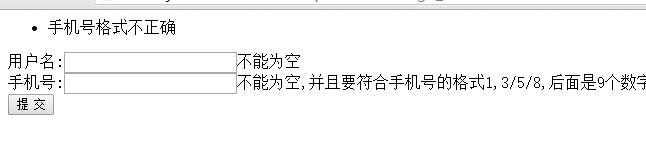
</package>

## 4.开始校验重写validate 这样会校验所有的与execute方法签名相同的方法，当某个方法校验失败，

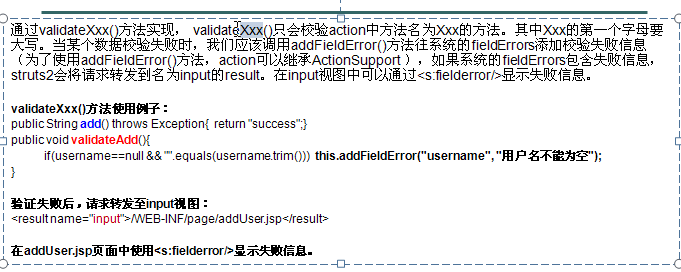


## 5测试





# 2.对action指定方法进行校验



## 1.只需要重写validateXxx（）方法，Xxx方法名字大写，比如我们只是校验update方法

//会对update()方法校验

**public** **void** validateUpdate() {

**if**(**this**.username==**null** || "".equals(**this**.username.trim())){

**this**.addFieldError("username", "用户名不能为空");

}

**if**(**this**.mobile==**null** || "".equals(**this**.mobile.trim())){

**this**.addFieldError("mobile", "手机号不能为空");

}**else**{

**if**(!Pattern.*compile*("^1[358]\\d{9}$").matcher(**this**.mobile).matches()){

**this**.addFieldError("mobile", "手机号格式不正确");

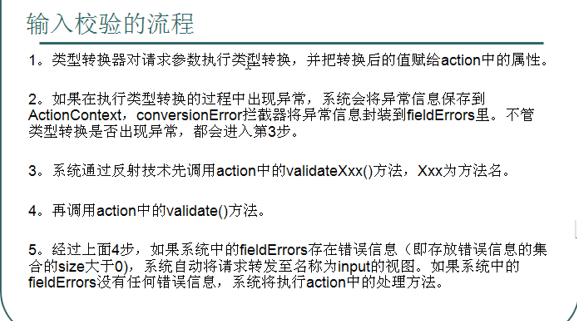
}

}

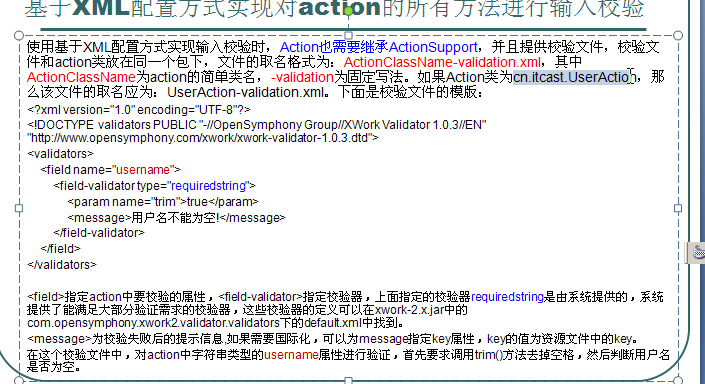
}

# 3.输入校验的流程

## 1.先会调用指定的方法校验



# 4. XML配置方式实现对action的所有方法进行校验



## 1.action也需要继承

**public** **class** PersonAction **extends** ActionSupport{

**private** String username;

**private** String mobile;

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getMobile() {

**return** mobile; }

**public** **void** setMobile(String mobile) {

**this**.mobile = mobile; }

**public** String update(){

ActionContext.*getContext*().put("message", "更新成功");

**return** "message";

}

**public** String save(){

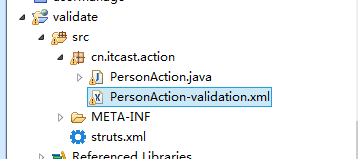
ActionContext.*getContext*().put("message", "保存成功");

**return** "message";

}

}

## 2.然后配置PersonAction-validation.xml，用到了*requiredstring校验器，这个是系统提供的，字符串trim是为校验器里面的注入值*



<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE validators PUBLIC "-//OpenSymphony Group//XWork Validator 1.0.3//EN" "http://www.opensymphony.com/xwork/xwork-validator-1.0.3.dtd">

<validators>

<field name=*"username"*>

<field-validator type=*"requiredstring"*> *要求是字符串*

<param name=*"trim"*>true</param>

<message>用户名不能为空!</message>

</field-validator>

</field>

<field name=*"mobile"*>

<field-validator type=*"requiredstring"*>

<message>手机号不能为空!</message>

</field-validator>**regex 要求符合格式**

<field-validator type=*"regex"*>

<param name=*"expression"*><![CDATA[^1[358]\d{9}$]]></param>

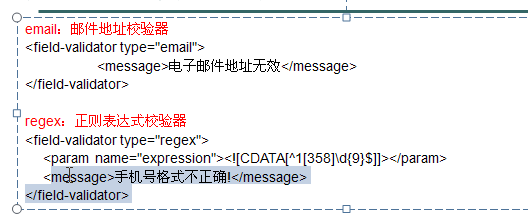
<message>手机号格式不正确!</message>

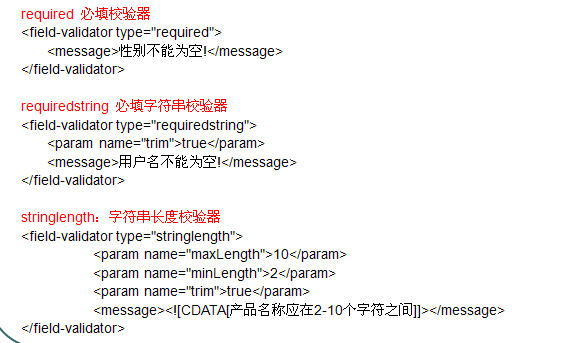
</field-validator>

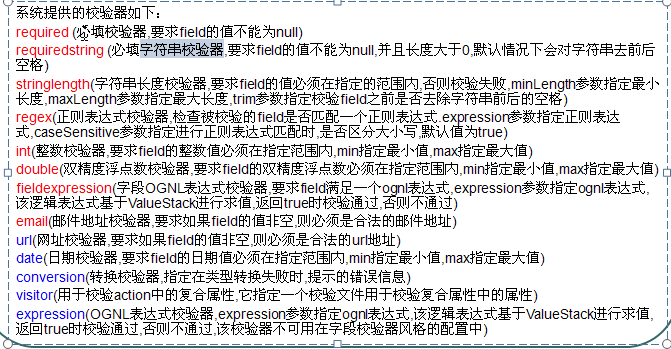
</field>

</validators>

## 3.各种校验器



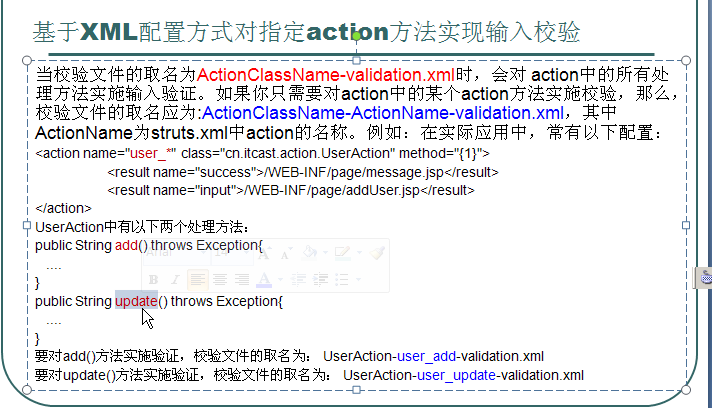




## 4.开始测试吧

验证成功

# 5.xml配置指定的校验



## 1.Struts中的配置文件为

<package name=*"person"* namespace=*"/person"* extends=*"struts-default"*>

<action name=*"manage\_\*"* class=*"cn.itcast.action.PersonAction"* method=*"{1}"*>

<!-- 校验错误会到input的界面 -->

<result name=*"input"*>/index.jsp</result>

<result name=*"message"*>/WEB-INF/page/message.jsp</result>

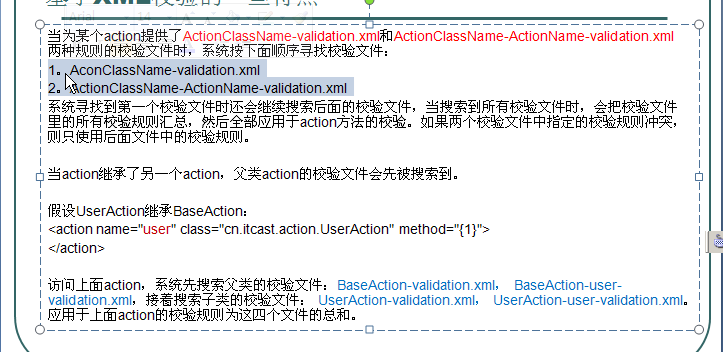
</action>

</package>

## 1.将校验文件的名字改成 与action的名称相同而且方法相同，先找包，然后找包下面的action

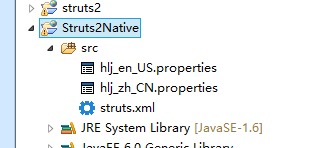
PersonAction-manage\_update-validation.xml

# 6.有多个校验文件的时候



# Struts 2的国际化

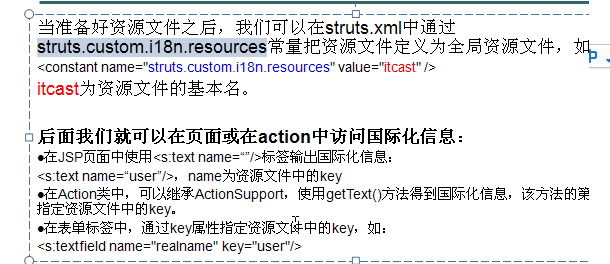
## 1.为应用添加两个资源文件



## welcome=欢迎来到中国

**welcome=welcome to china**

## 2.然后定义为全局范围内的资源文件



## 3.到了html中

<%@ taglib uri=*"/struts-tags"* prefix=*"s"*  %>

## 4.要是使用action访问资源文件可以 getTest()

