

Struts 2 进阶与提高

单元描述：

Struts 2 完全颠覆了 Web 编程的传统,在 Struts 2 的 Action 类中,完全去掉了 request、response 等 Servlet API。该框架提供了一整套用于简化 JSP 编程的标签,开发者只需要稍作配置就可以实现各种常用效果;该框架可以自动完成数据转换、赋值,这些对 Web 编程是革命性的转变;Struts 2 同样提供配置式的数据校验功能。本单元将重点介绍该框架在国际化、标签库、数据校验等方面的技术。

单元目标：

- 了解 Struts 2 标签库的配置和使用;
- 了解 Struts 2 在国际化方面的实现;
- 熟悉 Struts 2 数据转换和赋值的相关技术;
- 掌握 Struts 2 数据校验的功能。

5.1 任务1 引入性案例

任务描述：使用已经学过的有关 Struts 2 框架的知识,完成用户注册系统的开发。

任务目标：本案例的主要任务就是在当前知识体系结构的基础上完成用户注册模块的设计和实现,并分析不足之处。

5.1.1 案例分析

一个好的注册模块应该提供完备的数据验证功能。数据验证包括数据格式验证和逻辑验证,其中数据格式验证体现为数据格式的合法性,例如,在填写表单时要求密码不能为空、电话号码必须为 8~15 位整数、电子邮件地址的合法性等。逻辑验证一般体现为逻辑上的有效性,例如,判断用户注册的用户名是否重复等。

现在根据已学过的 Struts 2 的相关技术知识,完成用户登录模块。

工程名为 RegisterStruts2,工程文件结构图如图 5-1 所示。

用户注册页面如图 5-2 所示;用户名、密码、邮箱为空时的提示性信息如图 5-3 所示;年龄、电话号码和邮箱不合法的提示性信息如图 5-4 所示;用户已存在的提示性信息如图 5-5 所示;成功提交相关信息后进入欢迎页面,如图 5-6 所示。

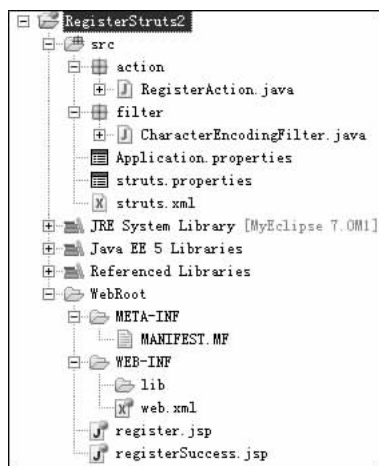


图 5-1 工程文件结构图



图 5-2 用户注册页面



图 5-3 用户名、密码、邮箱不能为空

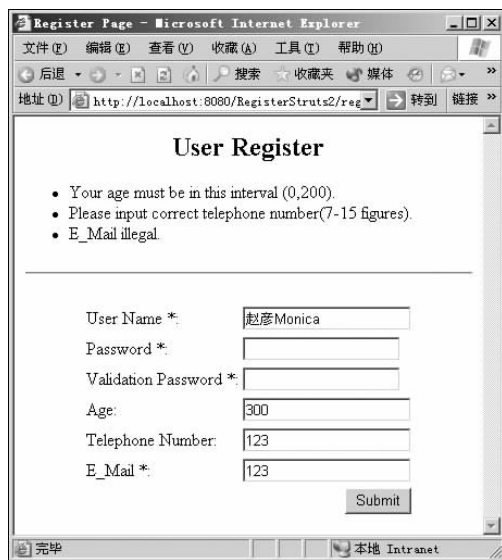


图 5-4 年龄、电话号码、邮箱不合法



图 5-5 用户名已存在



图 5-6 欢迎页面

5.1.2 设计步骤

- (1) 创建用户注册模块的视图界面: register.jsp、registerSuccess.jsp。
- (2) 创建资源信息文件: Application.properties。
- (3) 创建 Action: RegisterAction.java。
- (4) 为解决中文乱码问题, 创建过滤器 Servlet: CharacterEncodingFilter.java。
- (5) 创建配置文件 web.xml。
- (6) 配置 struts.xml。
- (7) 配置 struts.properties。
- (8) 发布工程。

5.1.3 具体实现

- (1) 创建用户注册模块的视图页面: register.jsp、registerSuccess.jsp。
register.jsp 的代码如下:

```
<%@page contentType="text/html; charset=UTF-8" language="java"%>
<%@taglib uri="/struts-tags" prefix="struts"%>
<html>
    <head><title><struts:text name="register.jsp.title"/></title></head>
    <body>
        <struts:text name="register.jsp.heading"/>
        <struts:fielderror/>
        <hr>
        <center>
            <struts:form action="register.action" method="post">
                <struts:textfield name="userName" key="register.jsp.label.username"/>
                <struts:password name="psw1" key="register.jsp.label.psw1"/>
                <struts:password name="psw2" key="register.jsp.label.psw2"/>
                <struts:textfield name="age" key="register.jsp.label.age"/>
                <struts:textfield name="telNum" key="register.jsp.label.telnum"/>
                <struts:textfield name="email" key="register.jsp.label.email"/>
                <struts:submit key="register.jsp.button.submit"/>
            </struts:form>
        </center>
    </body>
</html>
```

- registerSuccess.jsp 的代码如下:

```
<%@page language="java" pageEncoding="UTF-8"%>
<%@taglib uri="/struts-tags" prefix="struts"%>
<html lang="true">
```

```

<head><title><struts:text name="registerSuccess.jsp.title"/></title></head>
<body>
    <center><struts:text name="registerSuccess.jsp.heading"/></center>
    <hr>
    <struts:text name="registerSuccess.jsp.label.welcome"/>
    <struts:property value="userName"/>
    <struts:fielderror/>
</body>
</html>

```

(2) 创建资源信息文件：Application.properties。

该文件的内容与第4单元的进阶式案例(LoginStruts 2工程)的Application.properties文件类似,在这里不再重复给出。

(3) 创建 Action: RegisterAction.java。代码如下:

```

package action;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import com.opensymphony.xwork2.ActionSupport;
public class RegisterAction extends ActionSupport{
    private static final long serialVersionUID=1L;
    private String userName;
    private String psw1;
    private String psw2;
    private int age;
    private String telNum;
    private String email;
    //省略所有属性的 get、set 方法
    public String execute()throws Exception{
        if(getUserName().equals("ZhaoYan")){
            addFieldError("error0",getText("username.exist"));
            return SUCCESS;
        }
        else if(validator()){return SUCCESS;}
        else return ERROR;
    }
    private boolean validator(){
        boolean flag=true;
        if(getUserName()==null||getUserName().length()<1){ //判断用户名是否为空
            addFieldError("error1",getText("username.required"));
            flag=false;
        }
        if(getUserName().length()>20){ //用户名不能超过 20 个字符

```

```

        addFieldError("error2",getText("psw.maxlength"));
        flag=false;
    }
    if(getPsw1()==null||getPsw1().length()<1){ //判断密码是否为空
        addFieldError("error3",getText("psw.required"));
        flag=false;
    }
    if(!getPsw1().equals(getPsw2())){ //判断两次输入的密码是否一致
        addFieldError("error4",getText("errors.validwhen"));
        flag=false;
    }
    if(getAge()>200||getAge()<0){ //年龄应该是 0~200 之间的数字
        addFieldError("error5",getText("age.error"));
        flag=false;
    }
    String regTelNum="[0-9]{7,15}$";
    Pattern e=Pattern.compile(regTelNum);
    Matcher em=e.matcher(telNum);
    if(!(getTelNum().equals(""))&&!em.matches()){ //电话号码应该是 8~15 位的数字
        addFieldError("error6",getText("tel.error"));
        flag=false;
    }
    String regEmail="^([a-z0-9A-Z]+[-|\\.|?])+[a-z0-9A-Z]@([a-z0-9A-Z]+
        (-[a-z0-9A-Z]+)?\\.|)+[a-zA-Z]{2,}$"; //正则表达式
    Pattern p=Pattern.compile(regEmail); //Email 地址校验
    Matcher m=p.matcher(email);
    if(email==""){
        addFieldError("error7",getText("email.required"));
        flag=false;
    }
    if(email!=""&&!m.matches()){
        addFieldError("error7",getText("email.error"));
        flag=false;
    }
    return flag;
}
}

```

(4) 为解决中文乱码问题,创建过滤器 Servlet: CharacterEncodingFilter.java。

该文件和之前提到的 CharacterEncodingFilter.java 的代码相同,文件的配置也相同。

(5) 创建配置文件 web.xml。

该文件的配置与第4单元的进阶式案例中有关 web.xml 的配置相同,只是将默认欢迎页面改为 register.jsp。

(6) 配置 struts.xml。代码如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
    <package name="main" extends="struts-default">
        <action name="register" class="action.RegisterAction">
            <result name="error">/register.jsp</result>
            <result name="success">/registerSuccess.jsp</result>
        </action>
    </package>
</struts>
```

(7) 配置 struts.properties。

在该文件中,要配置相关的资源信息文件。在本工程中所有的文字信息都来源于资源信息文件,主要是为后面实现国际化做准备,让 Struts 2 框架识别资源配置文件的代码为“struts.custom.i18n.resources=Application”。该文件的代码如下:

```
struts.action.extension=action
struts.locale=en_GB
struts.custom.i18n.resources=Application
```

(8) 发布工程。

当所有代码输入完成后,启动 Tomcat,发布工程,然后在浏览器窗口查看运行效果。

5.1.4 Struts 2 工作流程

在使用 Struts 2 框架技术完成了两个案例之后,根据本案例,详细介绍一下 Struts 2 的工作流程。

Struts 2 放弃了 request、response 等 Servlet API,看上去更像普通的 Java 类。实际上,Struts 2 的 Action 属于被调用的方法。在调用 Action 的执行方法 execute()之前,Struts 2 会从 request 中获取参数,并通过 set()方法设置到 Action 属性中。整个 Struts 2 框架的工作流程如图 5-7 所示。

register.jsp 是一个表单页面,提交表单后,将数据提交到 register.action 中,这是一个 Struts 的 URI,将会被 Struts 2 配置在 web.xml 中的分发器 Filter 截获。分发器 Filter 转去查找 struts.xml,得知 register.action 对应的 Action 为 RegisterAction。

然后,Struts 2 将生成一个 RegisterAction 实例,并将提交上来的数据设置到该实

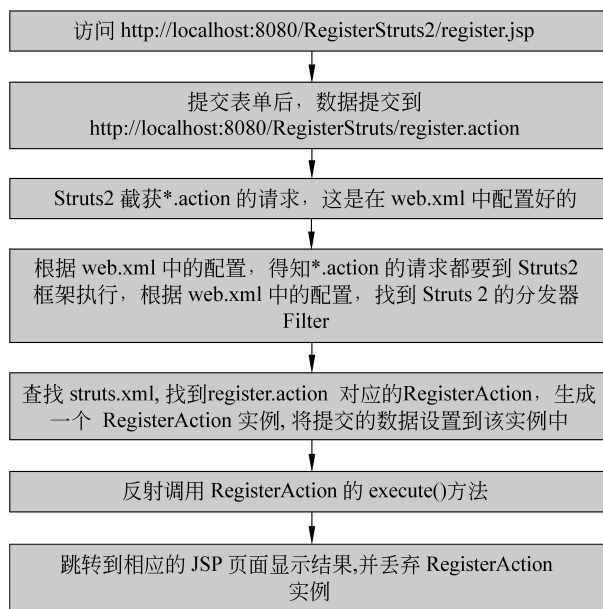


图 5-7 Struts 2 工作流程图

例中。

RegisterAction 中的 execute() 方法是 Struts 2 的主要方法, 主要的业务逻辑代码就在其中。本案例中如果输入的用户名为 ZhaoYan, 则显示该用户已存在的提示性信息; 如果用户输入无误, 则跳转到成功页面。而具体的页面跳转也已经在 struts.xml 中配置好了。

5.1.5 分析不足之处

通过两个 Struts 2 案例, 读者已经体会到了该框架的优势。尽管 Struts 2 避开了 Servlet API, 并且省去了严格的 ActionForm, 以及 JavaBean 的定义, 但是依据现有的知识来解决复杂的问题还是有一定难度的。在引入性案例设计之初, 尽管经过严格的需求分析, 还有很多功能难以实现。下面将一一点评。

(1) 一个优秀的注册程序不仅仅是输入用户名和密码, 在较为严格的系统中需要录入极为丰富的数据。为了方便用户录入, 就需要各种控件帮助用户完成录入工作, 如单选按钮、复选框、文件上传按钮、下拉框、日期选择器等。Struts 2 丰富的标签库是否可以完成该项工作呢?

(2) 国际化是一个产品应有的支持, 在 Struts 2 中是否也可以解决国际化问题呢?

(3) Struts 2 完全放弃了 ActionForm, 但是由于 Java 本身的缺陷, 比如时间的表示等, 前台表单传来的数据转去给后台 Java 处理时, 数据类型的转换往往会加大程序员的工作量, 那么 Struts 2 有没有更好的解决方案呢?

(4) 验证框架可以有效解决数据校验的问题, 那么 Struts 2 如何解决此类问题呢?

下面就带着这些问题来依次找到解决的答案,并改在进阶式案例中完成对用户注册模块的优化。

5.2 任务2 Struts 2 标签库

任务描述: Struts 2 提供了大量的标签,Struts 2 的标签库用于简化 JSP 编程,开发者只需要在标签中做少量配置,就可以实现各种常见效果。Struts 2 标签与 Action 联系比较紧密,使用标签后,Struts 2 会自动完成 JSP 层的显示数据、在 Action 层的采集数据等工作。

任务目标: Struts 2 提供了大量的标签,如日期选择器、树形结构、主题、模板等,同时 Struts 2 还提供了对 DWR 技术、AJAX 技术的支持,使 Struts 2 可以完成各种 AJAX 的效果。本任务的主要目标就是认识和了解 Struts 2 标签。

5.2.1 Struts 2 标签分类

早期的 Java Web 程序主要依靠 Java 代码控制输出,在这种情况下,JSP 页面中将嵌入大量的 Java 代码,使用大量的 if、else、for 等分割页面中静态和动态的部分,整个页面显得非常凌乱。自 JSP 1.1 版本以后,增加了标签的功能,从此 Java Web 框架技术中也提供了丰富的标签。

Struts 1 提供的标签已经可以在很大程度上简化 JSP 页面的编写,Struts 2 则提供了更为强大的标签库,使用 Struts 2 的标签库可以大大简化数据的输出,同时也可以实现更为绚丽的效果。

与 Struts 1 的标签库不同,Struts 2 的标签库提供的标签不依赖于任何表现层技术,也就是说,Struts 2 中的大部分标签可以在各种表现层技术中使用,这些表现层技术主要包括 JSP、Velocity、FreeMarker 等。当然,并不是所有的 Struts 2 标签都可以在所有的表现层技术中使用,存在极少数的标签,在某些表现层技术中使用时会受到一定的限制。

Struts 1 的标签库被分成了 5 类,分别是 html、bean、logic、nested 和 tiles。Struts 2 的标签库没有进行分类,所有的标签都被放到一个标签库中,用户指定前缀名,如 s、struts 等。

此外,Struts 1 的标签在默认情况下不支持表达式语言(EL),如果需要支持的话,要增加 struts-el.jar 和相应的 JSTL 库。Struts 2 在默认情况下支持 OGNL、JSTL、Groovy 和 Velocity 表达式。

Struts 2 的标签按照使用方式分为两类,一类是 UI 标签,另一类是非 UI 标签。所谓 UI 标签是指用于生成 HTML 元素的标签;所谓非 UI 标签主要是用于数据访问、逻辑控制等操作的标签。

对于 UI 标签而言,还可以分为表单标签、非表单标签、AJAX 标签。其中表单标签

是用于生成 HTML 页面的 form 标签,以及普通表单元素;非表单标签主要是用于生成 HTML 页面的其他 HTML 元素或显示某些信息的标签;AJAX 标签就是支持 AJAX 技术的 UI 标签。

对于非 UI 标签可以分为控制标签和数据标签。其中控制标签主要用于实现条件判断、循环控制;而数据标签主要用于输出 ValueStack 中的值,并完成国际化功能。Struts 2 标签库的分类如图 5-8 所示。

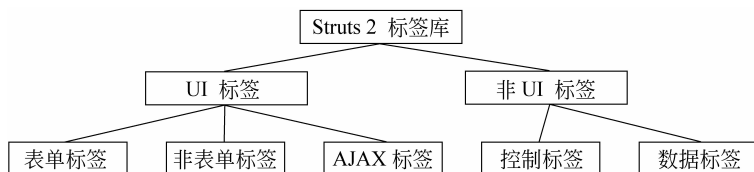


图 5-8 Struts 2 标签库的分类

5.2.2 表单标签

Struts 2 中所有的表单处理类都继承于 org.apache.Struts 2.components.UIBean 类,在该类中定义了一些通用属性,这些属性在所有的表单标签中都存在,感兴趣的读者可以阅读 UIBean 类的源代码。在 UIBean 类中定义的通用属性可分为模板相关属性、JavaScript 相关属性、tooltip 相关属性、通用属性 4 大类。表 5-1~表 5-4 分别对这四大类通用属性进行了详细描述。

表 5-1 模板相关属性

属 性	功 能 描 述
templDir	指定当前表单所用的模板文件目录
theme	指定当前表单所用的主题,如 simple、xhtml、ajax 等
template	指定当前表单所用的模板

表 5-2 JavaScript 相关属性

属 性	功 能 描 述
onclick	指定鼠标单击表单元素时触发的 JavaScript 函数或代码
ondblclick	指定鼠标双击表单元素时触发的 JavaScript 函数或代码
onmousedown	指定鼠标在表单元素上按下时触发的 JavaScript 函数或代码
onmouseup	指定鼠标在表单元素上抬起时触发的 JavaScript 函数或代码
onmouseover	指定鼠标在表单元素上悬停时触发的 JavaScript 函数或代码
onmouseout	指定鼠标移出表单元素时触发的 JavaScript 函数或代码

续表

属 性	功 能 描 述
onfocus	指定表单元素得到焦点时触发的 JavaScript 函数或代码
onblur	指定表单元素失去焦点时触发的 JavaScript 函数或代码
onkeypress	指定按下键盘某个键时触发的 JavaScript 函数或代码
onkeyup	指定松开键盘某个键时触发的 JavaScript 函数或代码
onkeydown	指定按下键盘某个键时触发的 JavaScript 函数或代码
onselect	该属性用于下拉列表框等表单元素,指定选中该元素时触发的 JavaScript 函数或代码
onchange	指定当前表单所用的模板

表 5-3 tooltip 相关属性

属 性	默 认 值	功 能 描 述
tooltip	无	设置组件的 tooltip 值
jsTooltipEnabled	false	设置是否支持 JavaScript
tooltipIcon	/struts/static/tooltip/tooltip.gif	设置 tooltip 图像的 URL
tooltipDelay	500	设置显示 tooltip 之间需要等待的时间(单位是毫秒)

表 5-4 通用属性

属 性	功 能 描 述
cssClass	设置 HTML 标签的 class 属性
cssStyle	设置 HTML 标签的 style 属性
title	设置 HTML 标签的 title 属性
disabled	设置 HTML 标签的 disabled 属性
label	设置表单元素的 label 属性,该属性值将根据当前表单元素不同的设置显示在页面的适当位置,系统会在该属性后自动加上冒号“:”
labelPosition	设置 label 属性值显示的位置,该属性有上面(top)和左边(left)两个值,默认值为左边
requiredposition	设置必填字段显示的必填标记(默认是星号“*”)的位置,该属性可设置的值为左边和右边(right)2 个值,默认值为右边
name	定义表单元素的 name 属性,该属性值需要和 Action 类的相应属性对应
required	指定当前表单元素是否为必填字段,如果是必填字段,会在该表单元素的相应位置显示必填标记(默认是星号“*”)
tabIndex	设置 HTML 标签的 tabIndex 属性
value	设置 HTML 标签的 value 属性