

学习目的与学习要求

学习目的：掌握 Struts 2 国际化的思路及开发流程、Struts 2 的异常机制，了解 Struts 2 的类型转换器，掌握 Struts 2 的校验框架。

学习要求：重点掌握 Struts 2 的国际化及验证框架，搭建流程，开发该知识点的实例。

本章主要内容

本章主要讲解 Struts 2 框架的高级技术，包括 Struts 2 的国际化、异常机制、类型转换和校验。其国际化包括国际化的思路及配置资源文件的种类，校验分为 validate 代码校验和校验框架。

前面讲解了 Struts 的基本原理，接下来介绍 Struts 的高级部分，包括 Struts 2 框架的国际化、异常处理、类型转换器和 Struts 2 验证框架等。首先讲解 Struts 对国际化的支持。

5.1 国际化支持

几年前，应用程序开发者能够考虑到仅支持他们本国的只使用一种语言（有时候是两种）和通常只有一种数量表现方式（如日期、数字、货币值）的应用。然而，基于 Web 技术的应用程序的爆炸性增长，以及将这些应用程序部署在 Internet 或其他被广泛访问的网络之上，已经在很多情况下使得国家的边界淡化到不可见。这种情况转变成为一种对于应用程序支持国际化（internationalization，经常被称作 i18n，因为 18 是字母 i 和字母 n 之间的字母个数）和本地化的需求。国际化是商业系统中不可或缺的一部分，所以无论您学习的是什么 Web 框架，它都是必须掌握的技能。

Struts 1.x 对国际化有很好的支持，它极大地简化了程序

员在做国际化时所需的工作。例如,如果要输出一条国际化信息,只在代码包中加入 FILE-NAME_xx_XX.properties(其中 FILE-NAME 为默认资源文件的文件名),然后在 struts-config.xml 中指明其路径,再在页面用<bean: message>标志输出即可。而 Struts 2.0 在原有的 Struts 1 简单、易用的基础上,将其做得更灵活、更强大。

1. Struts 2 的国际化概述

Struts 2 国际化建立在 Java 国际化的基础之上,也是通过提供不同国家/语言环境的消息资源,然后通过 ResourceBundle 加载指定 Locale 对应的资源文件,再取得该资源文件中指定 key 对应的消息——整个过程与 Java 程序的国际化完全相同,只要 Struts 2 框架对 Java 程序国际化进行了进一步封装,从而简化了应用程序的国际化。

1) 在 Struts 2 中加载全局资源文件

Struts 2 支持 4 种配置和访问资源文件的方法,包括:

- 使用全局的资源文件。
- 使用包范围的资源文件。
- 使用 Action 范围的资源文件。
- 使用<s:il8n>标志访问特定路径的 properties 文件。

其中最常用的是加载全局的国际化资源文件,至于其他几种方式,后面会讲解。加载全局的国际化资源文件的方式通过配置常量实现。不管在 struts.custom.xml 文件中配置常量,还是在 struts.properties 文件中配置常量,只配置 struts.custom.il8n.resources 常量即可。

配置 struts.custom.il8n.resources 常量时,该常量的值为全局国际化资源文件的 baseName。

假如系统需要加载的国际化资源文件的 baseName 为 properties/messageResource,则可以在 struts.properties 文件中指定如下一行:

```
Struts.custom.il8n.resources= properties.messageResource
```

或者,在 struts.xml 文件中配置如下的一个常量:

```
<!-- 定义资源文件的位置和类型 -->
<constant name="struts.custom.il8n.resources" value="properties/
messageResource"/>
```

通过这种方式加载国际化资源文件后,Struts 2 应用就可以在所有地方取出这些国际化资源文件了,包括 JSP 页面和 Action。

2) 访问国际化资源

Struts 2 既可以在 JSP 页面中通过标签输出国际化消息,也可以在 Action 类中输出国际化消息,不管采用哪种方式,Struts 2 都提供了支持,使用起来非常简单。

Struts 2 访问国际化消息主要有如下 3 种方式:

(1) 为了在 JSP 页面中输出国际化消息,可以使用 Struts 2 的<s:text.../>标签,该标签可以指定一个 name 属性,该属性指定了国际化资源文件中的 key。

(2) 为了在表元素里输出国际化信息,可以为该表单标签指定一个 key 属性,该 key 指定了国际化资源文件中的 key。

(3) 为了在 Action 类中访问国际化消息,可以使用 ActionSupport 类的 getText()方法,该方法可以接受一个 name 参数,该参数指定了国际化资源文件中的 key。

举例说明。首先是提供资源文件。

资源文件的内容就是 key-value 对

```
loginPage=loginPage
errorPage=errorPage
welcomePage=welcomePage
showBooksPage=showBooksPage
errorTip=sorry, you login error!!!
errorLink1=show books
errorLink2=here coming in GetBooksAction, but you did not login, so goto login path
welcomeTip=welcome, {0}, you login success!
welcomeLink=show books
loginTip=Login
user=username
password=password
submit=submit
showbookTip=show books
bookname=book name
username.required=username is required.
password.required=password is required.
```

上面的文件以 messageResource_en_US.properties 文件名保存,并将其保存在 WEB-INF/classes/properties 路径下,然后提供如下文件:

```
loginPage=登录页面
errorPage=错误页面
welcomePage=成功欢迎页面
showBooksPage=图书展现页面
errorTip=对不起,您登录失败!!!
errorLink1=展现图书
errorLink2=(此处应进入 GetBooksAction,但是检测到未登录,所以跳转到 login 的 path)
welcomeTip=欢迎, {0}, 您已经登录成功!
welcomeLink=展现书籍
loginTip=用户登录
user=用户名
password=密码
submit=提交
showbookTip=图书展示
bookname=书名
username.required=用户名不能为空
password.required=密码不能为空
```

使用 native2ascii 工具处理上面的资源文件,并命名为 messageResource_zh_CN.properties 保存在 WEB-INF/classes/properties 路径下,具体内容如下:

```
loginPage=\u767b\u9646\u9875\u9762
errorPage=\u9519\u8bef\u9875\u9762
welcomePage=\u6210\u529f\u6b22\u8fce\u9875\u9762
```

```

showBooksPage=\u56fe\u4e66\u5c55\u73b0\u9875\u9762
errorTip=\u5bf9\u4e0d\u8d77\u54c4\u60a8\u767b\u9646\u5931\u8d25\u54c4\u54c4\u54c4
errorLink1=\u5c55\u73b0\u56fe\u4e66
errorLink2=\u54c4\u6b64\u5904\u5e94\u8fdb\u5165GetBooksAction\u54c4\u4f46\u662f\u
u68c0\u6d4b\u5230\u672a\u767b\u9646\u6240\u4ee5\u8df3\u8f6c\u5230login\u7684path
\u54c4
welcomeTip=\u6b22\u8fce\u54c4, \u60a8\u5df2\u7ecf\u767b\u9646\u6210\u529f\u54c4
welcomeLink=\u5c55\u73b0\u4e66\u7c4d
loginTip=\u7528\u6237\u767b\u9646
user=\u7528\u6237\u540d
password=\u5bc6\u7801
submit=\u63d0\u4ea4
showbookTip=\u56fe\u4e66\u5c55\u793a
bookname=\u4e66\u540d
username.required=\u7528\u6237\u540d\u4e0d\u80fd\u4e3a\u7a7a
password.required=\u5bc6\u7801\u4e0d\u80fd\u4e3a\u7a7a

```

提供上面两份资源文件后,系统会根据浏览者所在的区域加载对应的语言资源文件。
下面是登录页面代码:

```

<%@ page language="java" contentType="text/html; charset=utf-8"%>
<!--导入 struts2 标签-->
<%@ taglib uri="/struts-tags" prefix="s"%>
<html>
<head>
<!--使用 s:text 标签输出国际化消息-->
<title><s:text name="loginPage"/></title>
</head>
<body>
<h3><s:text name="loginTip"/></h3>
<!--在表单元素中使用 key 指定国际化消息的 key-->
<s:form action="Login" method="post">
  <s:textfield name="username" key="user"/>
  <s:password name="password" key="password"/>
  <s:submit name="submit" key="submit" />
</s:form>
</body>
</html>

```

上面的 JSP 页面中使用<s:text.../>标签直接输出国际化信息,也通过在表单元素中指定 key 属性输出国际化信息。通过这种方式就可以完成 JSP 页面中普通文本、表单元素标签的国际化。

如果在简体中文环境下,浏览该页面将看到如图 5-1 所示的页面。

如果在控制面板中修改语言/区域,将机器的语言/区域环境修改成美国英语环境,再次浏览该页面,将看到如图 5-2 所示的页面。



图 5-1 登录页面



图 5-2 修改后的登录页面

如果为了在 Action 中访问国际化消息，则在 Action 类中调用 ActionSupport 类的 `getText()` 方法，就能够取得国际化资源文件中的国际化消息。通过这种方式，即使 Action 需要设置在下一个页面显示的信息，也无须直接设置字符串常量，而是使用国际化消息的 key 输出，从而实现程序的国际化。

下面是本示例应用中 Action 类的代码：

```
package com.ascent.action;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
@SuppressWarnings("serial")
public class LoginAction extends ActionSupport {
    private String username;
    private String password;
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    @SuppressWarnings("unchecked")
    public String execute() {
        if (getUsername().equals("ascent") && getPassword().equals("ascent")) {
            ActionContext.getContext().getSession().put("user", this.getUsername());
            return SUCCESS;
        }
        return ERROR;
    }
    //完成输入校验需要重写的 validate() 方法(读取资源文件 getText(String str))
    public void validate() {
        //调用 getText() 方法取出国际化消息
        if (getUsername() == null || "".equals(this.getUsername().trim())) {
```

```

        this.addFieldError("username", this.getText("username.required"));
    }
    if(this.getPassword()==null||"".equals(this.getPassword().trim())){
        this.addFieldError("password", this.getText("password.required"));
    }
}
}
}

```

注：这里使用了数据验证技术，为此需要重写 validate() 方法。如果用户登录时没有填写用户名或密码，系统就会报错。在简体中文环境下，提示消息分别为：用户名不能为空；密码不能为空。如果在控制面板中修改语言/区域，将机器的语言/区域环境修改成美国英语环境，那么提示消息分别变为 username is required; password is required。这样，也就提供了国际化的支持。

关于数据验证的详细内容，稍后会讲解，这里不赘述了。

2. 参数化国际化字符串

许多情况下，需要动态地为国际化字符插入一些参数，在 Struts 2.0 中可以方便地做到这一点。

如果需要在 JSP 页面中填充国际化消息里的占位符，则可以通过在 <s:text.../> 标签中使用多个 <s:param.../> 标签填充消息中的占位符。第一个 <s:param.../> 标签指定第一个占位符值，第二个 <s:param.../> 标签指定第二个占位符值，以此类推。

如果需要在 Action 中填充国际化消息里的占位符，则可以通过在调用 getText() 方法时使用 getText(String aTextName, List args) 或 getText(String key, String[] args) 方法填充占位符。该方法的第二个参数既可以是一个字符串数组，也可以是字符串组成的 List 对象。

在上述实例中，资源文件中有如下国际化消息：

```

#带占位符的国际化消息
welcomeTip=欢迎, {0}, 您已经登录成功!

```

为了在 Action 类中输出占位符的消息，我们在 Action 类中调用 ActionSupport 类的 getText() 方法，调用该方法时，传入用于填充占位符的参数值。访问该带占位符消息的 Action 类如下：

```

package com.ascent.struts2.action;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
@SuppressWarnings("serial")
public class LoginAction extends ActionSupport {
    private String username;
    private String password;
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}

```

```

    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    @SuppressWarnings("unchecked")
    public String execute() {
        if(getUsername().equals("ascent") && getPassword().equals("ascent")) {
            //调用 getText() 方法取出国际化消息,使用字符串数组传入占位符的参数值
            (request 范围)
            ActionContext.getContext().put("user", this.getText("welcomeTip", new
            String[]{this.getUsername()}));
            return SUCCESS;
        }
        return ERROR;
    }
}

```

通过上面的带参数的 `getText()` 方法,就可以为国际化消息的占位符传入参数了。

为了在 JSP 页面中输出带两个占位符的国际化消息,只为 `<s:text.../>` 标签指定 `<s:param.../>` 子标签即可。下面是 `welcome.jsp` 页面的代码:

```

<%@ page language="java" import="java.util.*" pageEncoding="utf-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<%
String path = request.getContextPath();
String basePath = request.getScheme() + "://" + request.getServerName() + ":" +
request.getServerPort() + path + "/";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<base href="<%=basePath%>">
<title><s:text name="welcomePage"/></title>
</head>
<body>
<!--使用 s:text 标签输出 welcomeTip 对应的国际化消息-->
<s:text name="welcomeTip">
<!--使用 s:param 为国际化消息的占位符传入参数-->
<s:param><s:property value="username"/></s:param>
</s:text>
<br><br>
<!-- 输出 request 范围内的 user 值(资源文件取值)-->
request:${requestScope.user}
<br>
<a href="getBooks.action"><s:text name="welcomeLink"/></a>

```

```

</body>
</html>

```

上面的页面使用 `${requestScope.user}` 输出的是 Action 类中取出的国际化消息, 而通过 `<s:text.../>` 标签取出的是另一个国际化消息, 且使用 `<param.../>` 标签为该国际化消息的占位符指定了占位符值。当以 ascent 用户名登录成功后, 结果如图 5-3 所示。

如果美国英语语言环境下用户通过登录页面登录成功, 进入 welcome.jsp 页面, 将看到如图 5-4 所示的页面。

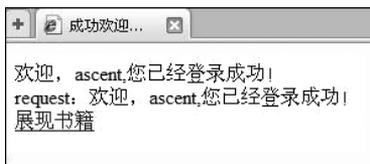


图 5-3 中文欢迎页面

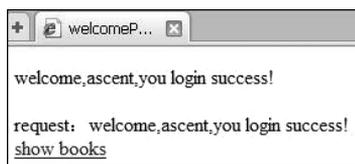


图 5-4 英文欢迎页面

登录失败 error.jsp 页面代码如下:

```

<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<!-- 导入 struts2 标签-->
<%@ taglib uri="/struts-tags" prefix="s"%>
<%
String path = request.getContextPath();
String basePath = request.getScheme()+ "://" + request.getServerName() + ":" +
request.getServerPort() + path + "/";
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<base href="<%=basePath%>">
<!-- 使用 s:text 标签输出国际化消息-->
<title><s:text name="errorPage"/></title>
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1, keyword2, keyword3">
<meta http-equiv="description" content="This is my page">
<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->
</head>
<body>
<s:text name="errorTip"></s:text><br>
<a href="login.jsp"><s:text name="errorLink1"></s:text></a>
<s:text name="errorLink2"></s:text>
</body>
</html>

```

用户名或密码错误时,中文和英文环境下的页面分别如图 5-5 和图 5-6 所示。



图 5-5 登录失败中文页面



图 5-6 登录失败英文页面

Struts 的配置文件 struts.xml 如下:

```
<? xml version="1.0" encoding="GBK"? >
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
  <constant name="struts.custom.i18n.resources"
value="properties/messageResource"/>
  <package name="struts2" extends="struts-default">
    <!-- 全局国际化测试类 -->
    <action name="login" class="com.ascent.action.LoginAction">
      <result>/welcome.jsp</result>
      <result name="error">/error.jsp</result>
      <result name="input">/login.jsp</result>
    </action>
  </package>
</struts>
```

3. 加载其他资源文件的方式

前面介绍了 Struts 2 中加载国际化资源最常用的方式,除此之外,Struts 2 还提供了多种方式加载国际资源文件,包括指定包范围资源文件、Action 范围资源文件,以及临时指定资源文件等。

1) 包范围资源文件

对于一个大型应用而言,国际化资源文件的管理也是一个复杂的工程,因为整个应用中有大量内容需要实现国际化,如果把国际化资源都放在同一个全局文件里,这将是不可想象的事

情。为了更好地体现软件工程里“分而治之”的原则，Struts 2 允许针对不同模块、不同 Action 组织国际化资源文件。

为 Struts 2 应用指定包范围资源文件的方法是：在包的根路径下建立多个文件名为 `package_language_country.properties` 的文件，一旦建立了这个系列国际化资源文件，应用中处于该包下的所有 Action 都可以访问该资源文件。

例如，`com.ascent.action2` 包下的 Action 类：

```
package com.ascent.action2;
import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionSupport;
@SuppressWarnings("serial")
public class LoginAction2 extends ActionSupport{
    private String username;
    private String password;
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    @SuppressWarnings("unchecked")
    public String execute() {
        if(getUsername().equals("ascent") && getPassword().equals("ascent")) {
            //调包范围资源文件
            ActionContext.getContext().put("tip", this.getText("succTip"));
            return SUCCESS;
        }
        ActionContext.getContext().put("tip", this.getText("failTip"));
        return ERROR;
    }
}
```

接着提供两份资源文件：第一份资源文件为 `package_zh_CN.properties`，内容为

```
failTip=\u5305\u8303\u56f4\u6d88\u606f\u5bf9\u4e0d\u8d77\u56e0\u60a8\u4e0d\u80fd\u767b\u5f55\u547d
succTip=\u5305\u8303\u56f4\u6d88\u606f\u5bf9\u6b22\u8c28\u56e0\u60a8\u5df2\u7ecf\u767b\u5f55\u547d
```

它使用 `native2ascii` 工具处理以下内容：

```
failTip=包范围消息:对不起,您不能登录!
```