Evan
Only let oneself become strong enough, good enough, can afford the life that you want to.

struts.xml配置详解

标签： struts2.0 struts.xml配置

2016-05-13 19:16 128人阅读 评论(0) 收藏 举报

分类： web后台（12） Struts2（1）

struts.xml是我们在开发中利用率最高的文件，也是Struts2中最重要的配置文件。

一下分别介绍一下几个struts.xml中常用到的标签

1、<include>

利用include标签，可以将一个struts.xml配置文件分割成多个配置文件，然后在struts.xml中使用<include>标签引入其他配置文件。

比如一个网上购物程序，可以把用户配置、商品配置、订单配置分别放在3个配置文件user.xml、goods.xml和order.xml中，然后在struts.xml中将这3个配置文件引入：

struts.xml：

```
[html]
<span style="font-size:14px;"><?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>
    <include file="user.xml"/>
    <include file="goods.xml"/>
    <include file="order.xml"/>
</struts></span>
```

user.xml：

```
[html]
<span style="font-size:14px;"><?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="wwfy" extends="struts-default">
        <action name="login" class="wwfy.user.LoginAction">
            <!--省略Action其他配置-->
        </action>
        <action name="logout" class="wwfy.user.LogoutAction">
            <!--省略Action其他配置-->
        </action>
    </package>
</struts></span>
```

2、<constant>

在之前提到struts.properties配置文件的介绍中，我们曾经提到所有在struts.properties文件中定义的属性，都可以配置在struts.xml文件中。而在struts.xml中，是通过<constant>标签来进行配置的：

```
[html]
<span style="font-size:14px;"><?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>
    <!--设置开发模式-->
    <constant name="struts.devMode" value="true"/>
    <!--设置编码形式为GB2312-->
    <constant name="struts.i18n.encoding" value="GB2312"/>
    <!--省略其他配置信息-->
</struts></span>
```

3、<package>

1、包属性介绍

在Struts2框架中是通过包来管理action、result、interceptor、interceptor-stack等配置信息的。包属性如下：

属性 是否必要

描述

name 是

包名，作为其它包应用本包的标记

extends 否 设置本包继承其它包

namespace 否 设置包的命名空间

abstact 否 设置为抽象包

2、extends属性的详解

当一个包通过配置extends属性继承了另一个包的时候，该包将会继承父包中所有的配置，包括action、result、interceptor等。

由于包信息的获取是按照配置文件的先后顺序进行的，所以父包必须在子包之前被定义。

通常我们配置struts.xml的时候，都继承一个名为“struts-default.xml”的包，这是struts2中内置的包。

3、namespace的详解

namespace主要是针对大型项目中Action的管理，更重要的是解决Action重名问题，因为不在同一个命名空间的Action可以使用相同

1）如果使用命名空间则URL将改变

比如我们有一下配置文件

```
[html]
<span style="font-size:14px;"><package name="wwfy" extends="struts-default">
    <action name="login" class="wwfy.action.LoginAction">
        <result>/success.jsp</result>
    </action>
</package>
 </span>
```

则此配置下的Action的URL为http://localhost:8080/login.action

假如为这个包指定了命名空间

```
[html]
<span style="font-size:14px;"><package name="wwfy" extends="struts-default" namespace="/user">
    <action name="login" class="wwfy.action.LoginAction">
        <result>/success.jsp</result>
    </action>
</package>
 </span>
```

则此配置下的Action的URL为http://localhost:8080/user/login.action

2)默认命名空间

Struts2中如果没有为某个包指定命名空间,该包使用默认的命名空间,默认的命名空间总是""。

3）指定根命名空间

当设置了命名空间为“/”，即指定了包的命名空间为根命名空间时，此时所有根路径下的Action请求都会去这个包中查找对应的资源信息。

假若前例中路径为http://localhost:8080/login.action则所有http://localhost:8080/*.action都会到设置为根命名空间的包中寻找资源。

4、<action>与<result>

1、<action>属性介绍

属性名称 是否必须 功能描述

name 是 请求的Action名称

class 否 Action处理类对应具体路径

method 否 指定Action中的方法名

converter 否

指定Action使用的类型转换器

如果没有指定method则默认执行Action中的execute方法。

2、<result>属性介绍

属性名称 是否必须

功能描述

name 否

对应Action返回逻辑视图名称，默认为success

type 否

返回结果类型，默认为dispatcher

3、通配符的使用

随着result的增加，struts.xml文件也会随之变得越来越复杂。那么就可以使用通配符来简化配置：

例如下面这个案例：

Action为Test.java

```
[java]
<span style="font-size:14px;">public class Test {
    public String test1(){
        return "result1";
    }

    public String test2(){
        return "result2";
    }

    public String test3(){
        return "result3";
    }
}

</span>
```

struts.xml中配置为

```
[html]
<span style="font-size:14px;"><package name="wwfy" extends="struts-default">
    <action name="test*" class="wwfy.action.test{1}">
        <result name="result{1}">/result{1}.jsp</result>
    </action>
</package>

</span>
```

4、访问Action方法的另一种实现方式

在Struts2中如果要访问Action中的指定方法，还可以通过改变URL请求来实现，将原本的“Action名称.action”改为“Action名称！方法名称.action”在struts.xml中就不需要指定方法名了。

5、<exception-mapping>与<global-exception-mapping>

这两个标签都是用来配置发生异常时对应的视图信息的,只不过一个是Action范围的,一个是包范围的,当同一类型异常在两个范围都被配置时,Action范围的优先级要高于包范围的优先级.这两个标签包含的属性也是一样的:

属性名称 是否必须

功能描述

name 否

用来表示该异常配置信息

result 是 指定发生异常时显示的视图信息,这里要配置为逻辑视图

exception 是 指定异常类型

两个标签的示例代码为:

```
[html]
<span style="font-size:14px;"><?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="default" extends="struts-default">
        <global-exception-mappings>
            <exception-mapping result="逻辑视图" exception="异常类型"/>
        </global-exception-mappings>
        <action name="Action名称">
            <exception-mapping result="逻辑视图" exception="异常类型"/>
        </action>
    </package></struts></span>
```

6、<default-class-ref>

当我们在配置Action的时候，如果没有为某个Action指定具体的class值时，系统将自动引用<default-class-ref>标签中所指定的类。

中，系统默认的class为ActionSupport，该配置我们可以在xwork的核心包下的xwork-default.xml文件中找到。

有特殊需要时，可以手动指定默认的class

```java
<span style="font-size:14px;">package wwfy.action;

public class DefaultClassRef {
    public void execute(){
        System.out.println("默认class开始执行……");
    }
}
 </span>
```

在struts.xml中配置

```html
<span style="font-size:14px;"><?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="wwfy" extends="struts-default">
        <!-- 指定默认class为Test -->
        <default-class-ref class="wwfy.action.DefaultClassRef"/>
        <action name="test1">
            <result>/index.jsp</result>
        </action>
    </package>
</struts>

</span>
```

7、<default-action-ref>

如果在请求一个没有定义过的Action资源时，系统就会抛出404错误。这种错误不可避免，但这样的页面并不友好。我们可以使用<default-action-ref>

来指定一个默认的Action，如果系统没有找到指定的Action，就会指定来调用这个默认的Action。

```html
<span style="font-size:14px;"><?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="wwfy" extends="struts-default">

        <default-action-ref name="acctionError"></default-action-ref>
        <action name="acctionError">
            <result>/jsp/actionError.jsp</result>
        </action>
    </package>
</struts>

</span>
```

8、<default-interceptor-ref>

该标签用来设置整个包范围内所有Action所要应用的默认拦截器信息。事实上我们的包继承了struts-default包以后，使用的是Struts的默认设置。我们可以在struts-default.xml中找到相关配置：

```
[html]
<span style="font-size:14px;"><default-interceptor-ref name="defaultStack"/></span>
```

在实际开发过程中，如果我们有特殊的需求是可以改变默认拦截器配置的。当时一旦更改这个配置，“defaultStack”将不再被引用，需要手动最加。

9、<interceptors>

通过该标签可以向Struts2框架中注册拦截器或者拦截器栈，一般多用于自定义拦截器或拦截器栈的注册。该标签使用方法如下：

```
[html]
<span style="font-size:14px;"><interceptors>
    <interceptor name="拦截器名" class="拦截器类"/>
    <interceptor-stack name="拦截器栈名">
        <interceptor-ref name="拦截器名">
    </interceptor-stack>
</interceptors></span>
```

10、<interceptor-ref>

通过该标签可以为其所在的Action添加拦截器功能。当为某个Action单独添加拦截器功能后，<default-interceptor-ref>中所指定的拦截器将不再对这个Action起作用。

11、<global-results>

该标签用于设置包范围内的全局结果集。在多个Action返回相同逻辑视图的情况下，可以通过<global-results>标签统一配置这些物理视图所对应的逻辑视图。

```
[html]
<span style="font-size:14px;"><?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="wwfy" extends="struts-default">
        <global-results>
            <result name="test">/index.jsp</result>
        </global-results>
    </package>
</struts></span>
```