Struts Code Pieces – DynaValidatorForm

This tutorial explains the usage of the DynaValidatorForm using a small working example.

Generals

Author:
Sascha Wolski
Sebastian Hennebrueder
http://www.laliluna.de/tutorials.html Tutorials for Struts, EJB, xdoclet, JSF, JSP and eclipse.

Date:
February, 8th 2005

Development Tools
Eclipse 3.x

Dependencies
Struts 1.1
Jboss or Tomcat
PDF download: http://www.laliluna.de/download/struts-dynavalidator-form-en.pdf
Source download: http://www.laliluna.de/download/struts-dynavalidator-form-source.zip

Table of Content

The DynaValidatorForm class

The form bean DynaValidatorForm is the dynamic variant of the ValidatorForm and offers the possibility to validate properties based on validation rules. These rules are defined in XML files.

The form beans of DynaValidatorForm are created by Struts based on the definition you configure in the Struts config file. Below you can see an example:

```
<form-beans >
   <form-bean name="exampleForm"
type="org.apache.struts.validator.DynaValidatorForm">
      <form-property name="age" type="java.lang.Integer" />
      <form-property name="name" type="java.lang.String" />
   </form-bean>
</form-beans>
```

The Form Bean can be used in an Action. Below you can see an example ActionMapping.

Example:

```
<action attribute="exampleForm"
        input="/form/example.jsp"
        name="exampleForm"
        path="/example"
        scope="request"
        type="my.package.ExampleAction" />
```

Validation of Properties

The form bean DynaValidatorForm uses the Struts validation capabilities using validation rules defined in XML files. Struts offers a wide choice of rules, you can all find in the file validator-rules.xml.

You configure the rules for each property of a FormBean. These validations have to be written in the XML file (validation.xml)

Example validation file validation.xml:

```
<form-validation>
 <formset>
        <!-- validation mapping für example form -->
        <form name="exampleForm">
            <field
                 property="name"
               depends="required, minlength">
                 <arg0 key="exampleForm.name" />
                 <arg1 key="${var:minlength}" resource="false" />
                        <var>
                          <var-name>minlength</var-name>
                          <var-value>3</var-value>
                        </var>
            </field>
        </form>
    </formset>
</form-validation>
```

Initialization of properties

You can specify a default value for each property using the **initial** attribute in the <form-property> tag.

Example:

```
<form-beans>
  <form-bean name="exampleForm"
type="org.apache.struts.validator.DynaValidatorForm">
      <form-property name="age" type="java.lang.Integer" initial="23" />
      <form-property name="name" type="java.lang.String" initial="Adam" />
      </form-bean>
</form-beans>
```

Working example using the ValidatorForm Beans

Using a small working example we will show you the use of the DynaValidatorForm Bean.

Create the form bean (struts-config.xml)

Open the *struts-config.xml* and add a new form bean tag to the form beans area. Add two properties, name of type String and age of type Integer.

Specify default values:

Below you can see the example code.

```
<form-beans >
```

```
       <form-bean name="exampleForm"
type="org.apache.struts.validator.DynaValidatorForm">
          <form-property name="age" type="java.lang.Integer" initial="23" />
          <form-property name="name" type="java.lang.String" initial="Adam
Weisshaupt" />
       </form-bean>
</form-beans>
```

Create the Action class

Create the class *ExampleAction in the p*ackage de.laliluna.tutorial.dynavalidatorform.action*.*
The class extends the class Action.

Implement the method *execute(..)*.

Output the name and the age to the log.

The complete source code is shown below.

```
public class ExampleAction extends Action {

      public ActionForward execute(
            ActionMapping mapping,
            ActionForm form,
            HttpServletRequest request,
            HttpServletResponse response) {

            //ActionFormDynaValidatorForm zuweisen
            ExampleFormDynaValidatorForm exampleForm =
(ExampleFormDynaValidatorForm) form;

            //access the properties of the ActionForm

            System.out.println(exampleForm.get("name"));
            System.out.println(exampleForm.get("age"));

            return mapping.findForward("showExample");
      }

}
```

Create a JSP file

Create a JSP example.jsp in the directory *../WebRoot/form/* .

Below you can see the source code of the JSP file.

```
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html"%>

<html>
      <head>
            <title>JSP for exampleForm</title>
      </head>
      <body>
            <html:form action="/example">
                  <html:errors />
                  Name: <html:text property="name" /> <br>
                  Age: <html:text property="age" /> <br>
                  <html:submit value="Send"/>
            </html:form>
      </body>
</html>
```

Configure the Action (struts-config.xml)

Add an action mapping in the struts-config.xml. Add the form bean *exampleForm* to the action and create a forward to the *example.jsp*.

name specifies the action of the form bean.

Type is the path to our action class, *ExampleAction*.

Input specifies the JSP, you are forwarded to when an error occurred in the validation phase.

<forward ...> is the forward to our *example.jsp*.

```
<action-mappings>
      <action
          attribute="exampleForm"
          input="/form/example.jsp"
          name="exampleForm"
          path="/example"
          scope="request"
          type="de.laliluna.tutorial.dynavalidatorform.action.ExampleAction">

              <forward name="showExample" path="/form/example.jsp" />

      </action>
</action-mappings>
```

Validating properties with XML validation rules

To validate the user input, if a name's length is greater than 3 character or the age is between 0 and 150, you have to configure this validations in an XML file.

Create the XML file *validation.xml* in the directory
/WebRoot/WEB-INF/.

<form name=".."> defines the Form Bean to which the validations are applied.

<field property=".."> defines a property of a form bean. The attribute *depends* configures the used rule from the Struts rule set. (All rules are defined in the validator-rules.xml).

<arg0 key=".."> defines a parameter which is passed to the error message. In the error message for intRange, there is one parameter expected. (more informations at MessageResouce).

<var-name> sets the name of the variable used in the validation rule and *<var-value>* the value of the variable.

Create the following validations for the form bean property:

```
<form-validation>
<formset>
      <!-- validation mapping für example form -->
      <form name="exampleForm">
         <field
           property="name"
           depends="required, minlength">
                <arg0 key="exampleForm.name" />
                <arg1 key="${var:minlength}" resource="false" />
                    <var>
                            <var-name>minlength</var-name>
                            <var-value>3</var-value>
                    </var>
         </field>
         <field
           property="age"
           depends="intRange">
                <arg0 key="exampleForm.age"/>
```

```
                    <arg1 name="intRange" key="${var:min}" resource="false" />
                    <arg2 name="intRange" key="${var:max}" resource="false" />
                        <var>
                                <var-name>min</var-name>
                                <var-value>1</var-value>
                        </var>
                        <var>
                                <var-name>max</var-name>
                                <var-value>150</var-value>
                        </var>
                </field>
        </form>
    </formset>
</form-validation>
```

Configure the ValidatorPlugins in the Struts Config file

In order to use the Struts-Validator you must add the ValidatorPlugin in the Struts Config. Otherwise Struts does not know your validation files and will not use them.

Open the struts-config.xml and add the following properties to the end of the struts config file.into the tag *<struts-config>* .

```
<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
    <set-property
        property="pathnames"
        value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml"/>
</plug-in>
```

Create a Message Resource file

The Message Resource file is needed for the output of the error messages, we used in the execute method.

Create a new file named *ApplicationResources.properties* in the package *de.laliluna.tutorial.dynavalidatorform.*

You can find more information about message resource files in our Message Resource tutorial. http://www.laliluna.de/struts-message-resources-tutorial.html

Add the following to the file:

```
errors.suffix=<br>
# -- default error messages for struts validator
errors.required='{0}' is required.
errors.minlength='{0}' can not be less than {1} characters.
errors.range='{0}' is not in the range {1} through {2}.
# -- field names
exampleForm.name=Name
exampleForm.age=Age
```

Open the *struts-config.xml* and add the following lines to configure your resource file.

```
<message-resources
parameter="de.laliluna.tutorial.dynavalidatorform.ApplicationResources" />
```

Test your example

We have finished our example application. Test the example by calling

http://localhost:8080/DynaValidatorForm/example.do