

# Struts Code Pieces – LazyValidatorForm

This tutorial explains the usage of the form bean LazyValidatorForm using a small example application.

## Generals

### Author:

Sascha Wolski

Sebastian Hennebrueder

<http://www.laliluna.de/tutorials.html> Tutorials for Struts, EJB, xdoclet, JSF, JSP and eclipse.

### Date:

February, 8<sup>th</sup> 2005

## Development Tools

Eclipse 3.x

## Dependencies

Struts 1.1

Jboss, Tomcat, Jetty etc

PDF download: <http://www.laliluna.de/download/struts-lazyvalidator-form-tutorial-en.pdf>

Source download: <http://www.laliluna.de/download/struts-lazyvalidator-form-tutorial-source.zip>

## Table of Content

Struts Code Peaces – LazyValidatorForm.....	1
Generals.....	1
Die LazyValidatorForm Klasse.....	1
Validation of Properties.....	2
Initialization of properties.....	2
Working example using the ValidatorForm Beans.....	2
Create the form bean (struts-config.xml).....	2
Create the Action class.....	3
Create a JSP file.....	3
Configure the Action (struts-config.xml).....	3
Validating properties with XML validation rules.....	4
Configure the ValidatorPlugins in the Struts Config file.....	5
Create a Message Resource file.....	5
Test your example.....	5

## Die LazyValidatorForm Klasse

The LazyValidatorForm is a dynamic variant of the form bean. It does not need a java class implementation but is created dynamically by Struts. You configure the form bean in the Struts config file.

The special advantage of this form bean is that you do not have to specify any properties. The properties are created dynamically once a form is posted.

Example of a LazyValidatorForm form bean declaration:

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

## Working example using the ValidatorForm Beans

Using a small working example we will show you the use of the `LazyValidatorForm` form beans.

### Create the form bean (`struts-config.xml`)

Open the `struts-config.xml` and add a new form bean tag of type `LazyValidatorForm` to the form beans area.

We will not declare any properties to show the capabilities of a lazy form bean to create them.

Below you can see the declaration of the form bean in the Struts configuration file.

```
<form-beans >
  <form-bean name="exampleForm"
```

```
type="org.apache.struts.validator.LazyValidatorForm" />
</form-beans>
```

## Create the Action class

Create the class *ExampleAction* in the package *de.laliluna.tutorial.lazyvalidatorform.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) {

        //LazyValidatorForm zuweisen
        LazyValidatorForm exampleForm = (LazyValidatorForm) form;

        //Zugriff auf Eigenschaften der LazyValidatorForm
        //Klasse innerhalb der Action Klasse
        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.

```
<%@ page language="java"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean"%>
<%@ 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.lazyvalidatorform.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 *MessageResource*).

`<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="required, intRange, integer">
      <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>
```

```
        </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.lazyvalidatorform*.

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}.
errors.integer={0} must be an integer.
# -- 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.lazyvalidatorform.ApplicationResources" />
```

## Test your example

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

<http://localhost:8080/LazyValidatorForm/example.do>