

# Struts Code Peaces – ActionForm

This tutorial explains the Struts form bean ActionForm 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 3.2.5 or Tomcat

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

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

## Table of Content

Struts Code Peaces – ActionForm.....	1
Generals.....	1
The ActionForm Klasse.....	1
Validation of properties.....	2
Initializing the properties of the ActionForm class.....	2
Working example of an ActionForm Bean.....	2
Create an ActionForm class.....	2
Create the Action class.....	3
Create a JSP file.....	3
Configure a FormBean (struts-config.xml).....	4
Configure the Action (struts-config.xml).....	4
Initializing the properties of the ActionForm class.....	4
Validate the properties in the actionForm class.....	5
Create a Message Resource file.....	5
Test your example.....	5

## The ActionForm Klasse

This form type is a normal java class extending the ActionForm class.

Example:

```
public class ExampleForm extends ActionForm { }
```

Once it is created, you have to specify a name for the FormBean in the struts configuration file

Example:

```
<form-beans >  
    <form-bean name="exampleForm" type="my.package.ExampleForm" />  
</form-beans>
```

The form bean can be used in an Struts action. Below there is an example of an ActionMapping

using our form bean.

Example:

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

## Validation of properties

You can implement a „validate“ method in the class `ActionForm`. In this method you can (but you must not) validate the properties. This method is called after a form is submitted, resetted and filled with the new values. You can validate fields, if the content is correct and whatever else. When you return a non empty `actionErrors` then struts will bring you back to the page you specified with the „input“ tag. On this page you can output your error messages.

Example:

```
public ActionErrors validate(
    ActionMapping mapping,
    HttpServletRequest request) {

    //create a new instance of actionErrors
    ActionErrors actionErrors = new ActionErrors();

    //Validate the text property
    if(text.length() == 0)
        actionErrors.add("example", new ActionMessage("Field should not be
empty!"));

    //Return the errors
    return actionErrors;
}
```

## Initializing the properties of the ActionForm class

Add an reset method to the `actionForm` class. This method is called by Struts when it initializes an `ActionForm`. You can define Default values to the form bean attributes in this method.

Example:

```
public void reset(ActionMapping mapping,
    HttpServletRequest request) {

    //Initialize the property
    text = "Hello World";
}
```

## Working example of an ActionForm Bean

Using a simple example we will show you now the usage of an `ActionForm`.

### Create an ActionForm class

Create a new class named `ExampleForm` in the package `de.laliluna.tutorials.actionform.form`.

The class should extend the class `ActionForm`.

Add two properties name and age.

Create getter and setter methods for each property.

The class should look like the following.

```
public class ExampleForm extends ActionForm {

    //properties
    private String name;
    private int age;

    //Getter and Setter methods
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

## Create the Action class

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

        //convert the ActionForm
        ExampleForm exampleForm = (ExampleForm) form;

        //access the properties and output them
        System.out.println(exampleForm.getName());
        System.out.println(exampleForm.getAge());

        return mapping.findForward("success");
    }
}
```

## 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 a FormBean (struts-config.xml)

Open the *struts-config.xml* and add a form bean declaration, between the *<form-beans>* tags, The attribute name defines the name of the FormBean, by which it can be used in action mappings.

*type* is the name of our class with the complete package name in front of it. In our case is the name *ExampleForm*.

```

<form-beans >
    <form-bean name="exampleForm"
type="de.laliluna.tutorial.actionform.form.ExampleForm" />
</form-beans>

```

## Configure the Action (struts-config.xml)

Add a 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.*

*<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.actionform.action.ExampleAction">

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

    </action>
</action-mappings>

```

## Initializing the properties of the ActionForm class

Add an reset method to the actionForm class. This method is called by Struts when it initializes an ActionForm. We will initialize our two fields.

```

public void reset(ActionMapping mapping,
    HttpServletRequest request) {

    //Initializing the properties
    name = "Adam Weisshaupt";
    age = 23;
}

```

## Validate the properties in the actionForm class

We will validate if the name is longer than three characters and if the age is greater than 0. As explained before, the validation is only possible in the action class.

The source code below is the *validate(..)* method of the *ExampleForm* class.

```
public ActionErrors validate(
    ActionMapping mapping,
    HttpServletRequest request) {

    //Create a new instance of ActionErrors
    ActionErrors actionErrors = new ActionErrors();

    //validate the properties of the DynaActionForm
    if(exampleForm.get("name").toString().length() < 3){
        actionErrors.add(ActionErrors.GLOBAL_ERROR, new ActionError("error.name"));
    }

    if(Integer.parseInt(exampleForm.get("age").toString()) < 1){
        actionErrors.add(ActionErrors.GLOBAL_ERROR, new ActionError("error.age"));
    }

    //return the actionErrors
    return actionErrors;
}
```

## 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.dynaactionform*.

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>
error.name=Name must have minimum 3 characters
error.age=Age must be greater then 0
```

Open the *struts-config.xml* and add the following lines.

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

## Test your example

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

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

(We expect a standard installation of JBOSS or Tomcat)