

ValidatorDispatchAction

up to [ValidatorFAQ](#)

How to use validator with dispatchAction

Answer(s)

if you want to use different validation rules with the validator you have to declare different action elements in your struts-config.xml file (using the same action class if you want). for that you have to use the (Dyna)ValidatorActionForm.

another means is to validate your form in the methods of your [DispatchAction](#) class as in this example:

struts-config.xml:

```
<action      path="/logon/action/logon"
             type="test.web.struts.action.logon.LogonAction"
             name="logonForm"
             scope="request"          <-- !!!
             validate="false"         <-- !!!
             input="/WEB-INF/jsp/logon/logon.jsp">
    <forward  name="success"   path="/index.jsp" />
</action>
```

[LogonAction.java](#):

```
ActionErrors errors = new ActionErrors();
errors = form.validate(mapping, request);

// Report any errors we have discovered back to the original form
if (!errors.isEmpty())
{
    saveErrors(request, errors);
    return new ActionForward(mapping.getInput());
}
```

<eliemedeiros@hotmail.com>

Or you could also write a thin wrapper around the validationframework to validate a single field at the time. For instance the following code carries out the validation of the form field at the same time it retrieves its value:

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import javax.servlet.http.*;

import org.apache.struts.action.*;
import org.apache.struts.validator.DynaValidatorForm;

public class CustomValidator {
```

[ActionErrors](#) formValidationErrors;

[DynaValidatorForm](#) form;

private static final Logger logger = Logger.getLogger(MosaicValidator.class);

```
/**
 *
 * Creates a new MosaicValidator object.
 * @param form parameter
 * @param mapping parameter
 * @param request parameter */
public MosaicValidator(DynaValidatorForm form, ActionMapping mapping, HttpServletRequest request) { this.form = form; formValidationErrors =
form.validate(mapping, request); }
/**
 * Get the value for a particular form field, and validate the value the user has entered against the Struts validation file
 * @param key parameter
 * @param errors parameter
```

- @return returnValue */
public final Object validate(String key, [ActionErrors](#) errors) { Iterator it = formValidationErrors.get(key);

```
while (it.hasNext()) {
```

```
ActionError e = (ActionError)it.next();  
errors.add(key, e);  
}
```

```
return form.get(key);
```

```
}
```

```
}
```

This can be called in the following manner from an action extending [DispatchAction](#):

```
public ActionForward viewForm (ActionMapping mapping, ActionForm form, HttpServletRequest request,  
HttpServletRequest response)
```

```
throws Exception {  
ActionErrors errors = new ActionErrors();  
MosaicValidator v=new MosaicValidator( (DynaValidatorForm)form, mapping, request);  
String formField = (String)v.validate("formField", errors);  
if (!errors.isEmpty()) {  
saveErrors(request, errors);
```

```
return (mapping.getInputForm() );
```

```
}  
return viewForm(mapping, form, request, response);  
}
```

Hope this helps someone.

</eliemedeiros@hotmail.com>

up to [ValidatorFAQ](#)