Struts 2

Struts 2 Validation

To avoid the wrong values, we need to perform validation on forms where user submits some values. For example, if user writes his/her email id as abc, we need to give error message to the user that the given email id is not correct. So that we can have only valuable informations.

There are three ways to perform validation in struts 2.

1) By Custom Validation Here, we must implement the Validateable interface (or extend ActionSupport class) and provide the implementation of validate method.

2) By Input Validation (built-in validators) Struts 2 provides a lot of predefined that can be used in struts 2 application to perform validation.

Struts 2 provides following bundled validators.

- requiredstring validator
- stringlength validator
- email validator
- date validator
- int validator
- double validator
- url validator
- regex validator

3) By Ajax Validation (built-in validators with ajax) If we don't want to refresh the page, we can use jsonValidation interceptor to perform validation with ajax.

Struts 2 Custom Validation

We can define our own validation logic (custom validation) in struts 2 by implementing the **Validateable** interface in the action class.

The **workflow interceptor** is used to get information about the error messages defined in the action class.

Workflow Interceptor

The **workflow interceptor** checks if there is any validation errors or not. It doesn't perform any validation.

It is applied when action class implements the Validateable interface. The **input** is the default parameter for this interceptor that determines the result to be invoked for the action or field error.

It is found in the defaultStack so we don't need to define it explicitly.

Validateabale interface

The **Validateable** interface must be implemented to perform validation logic in the action class. It contains only one method **validate()** that must be overridden in the action class to define the validation logic. Signature of the validate method is:

public void validate();

ValidationAware interface

The **ValidationAware** interface can accept the **field level** or **action class level** error messages. The field level messages are kept in Map and Action class level messages are kept in collection. It should be implemented by the action class to add any error message.

Steps to perform custom validation

The steps are as follows:

1. **create the form to get input from the user**
2. **Define the validation logic in action class by extending the ActionSupport class and overriding the validate method**
3. **Define result for the error message by the name input in struts.xml file**

Example to perform custom validation

In this example, we are creating 4 pages :

1. **index.jsp** for input from the user.

2. **RegisterAction.java** for defining the validation logic.
3. **struts.xml** for defining the result and action.
4. **welcome.jsp** for the view component.

1) Create index.jsp for input

This jsp page creates a form using struts UI tags. It receives name, password and email id from the user.

2) Create the action class

This action class inherits the ActionSupport class and overrides the validate method to define the validation logic.

RegisterAction.java

3) Define a input result in struts.xml

This xml file defines an extra result by the name input, that will be invoked if any error message is found in the action class.

4) Create view component

It is the simple jsp file displaying the information of the user.

welcome.jsp