Struts 2

Struts 2 params interceptor example

The **params interceptor** also known as parameters interceptor is used to set all parameters on the valuestack.

It is found in the default stack bydefault. So you don't need to specify it explicitely.

Internal working of params interceptor

It gets all parameters by calling the **getParameters() method of ActionContext** and sets it on the valuestack by calling the **setValue() method of ValueStack**.

Struts 2 defaultStack interceptors

The params interceptor is found in the default stack. You don't need to specify interceptors found in the default-stack. The defaultStack interceptors are as follows:

```
<interceptor-stack name="defaultStack">
        <interceptor-ref name="exception"/>
        <interceptor-ref name="alias"/>
        <interceptor-ref name="servletConfig"/>
        <interceptor-ref name="prepare"/>
        <interceptor-ref name="i18n"/>
        <interceptor-ref name="chain"/>
        <interceptor-ref name="debugging"/>
        <interceptor-ref name="profiling"/>
        <interceptor-ref name="scopedModelDriven"/>
```

```
      <interceptor-ref name="modelDriven"/>
      <interceptor-ref name="fileUpload"/>
      <interceptor-ref name="checkbox"/>
      <interceptor-ref name="staticParams"/>
      <interceptor-ref name="actionMappingParams"/>
      <interceptor-ref name="params">
       <param name="excludeParams">dojo\..*,^struts\..*</param>
      </interceptor-ref>
      <interceptor-ref name="conversionError"/>
      <interceptor-ref name="validation">
        <param name="excludeMethods">input,back,cancel,browse</param>
      </interceptor-ref>
      <interceptor-ref name="workflow">
        <param name="excludeMethods">input,back,cancel,browse</param>
      </interceptor-ref>
    </interceptor-stack>
```

Struts 2 execAndWait interceptor example

The **execAndWait interceptor** also known as execute and wait interceptor is used to display the intermediate result.

It is recommended to use for long running action.

It is not found in the default stack bydefault. So you need to specify it explicitely.

If you don't specify **"wait"** result, struts framework displays an intermediate result until your request is completed.

For the custom intermediate result, you need to define "wait" result in struts.xml file. In your page, you can display processing image etc. So, it is better to specify the custom result.

Example of execAndWait interceptor without wait result

Let's see the simple example of execAndWait interceptor without wait result. In such case, struts framework provides an intermediate result.

```
<action name="login" class="com.javatpoint.Login">
```

```
    <interceptor-ref name="params"/>
    <interceptor-ref name="execAndWait"/>
    <result name="success">login-success.jsp</result>
</action>
```

Example of execAndWait interceptor with wait result

Let's see the simple example of execAndWait interceptor with wait result. In such case, your intermediate page is invoked.

```
<action name="login" class="com.javatpoint.Login">
    <interceptor-ref name="params"/>
    <interceptor-ref name="execAndWait"/>
    <result name="success">login-success.jsp</result>
    <result name="wait">myintermediatepage.jsp</result>
</action>
```

myintermediatepage.jsp

Let's write the code for intermediate result. The s:url tag will forward the request to specified url.

It will be refreshed at 0.5 seconds.