

Creating stxx Actions

by Don Brown

A guide to stxx

Table of contents

1 The stxx Action.....	2
2 The XML Document.....	3

1. The stxx Action

First of all, when creating a new Action class, you will need to change the base class that your Action class extends. Previously, you would want to extend `org.apache.struts.action.Action`. Now, you will want to extend `com.oroاد.stxx.action.Action`. This change slightly introduces a new method called `saveDocument(HttpServletRequest, org.jdom.Document OR org.w3c.Document)`. This new method works in the same way as the Struts `saveErrors` method that you use when using the `ActionErrors` object. The document in this case is saved to the request object as an attribute. The object is later retrieved by the stxx transformer to create the output type.

In the code for the `perform` method of the Action class, you would populate the Document object with XML. This document object will then get retrieved later by the stxx transformer where it will be processed. This usually means an XSL file will transform it. Think of the document object as an XML data stream that is the result of running the Action's business logic.

```
public class ContactListExampleAction extends Action {
    public ActionForward perform(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response)
        throws IOException, ServletException {

        //create a new XML document for this Action with the root
        //element of "contactListExample"
        Document document =
            new Document(new Element("contactListExample"));

        //add some data to the XML document so that the Action
        //will produce XML in the form
        Element contactList = new Element("contactList");
        Element contact = new Element("contact");
        contact.addContent(new Element("display")
            .setText("Blow, Joe"));
        contact.addContent(new Element("email")
            .setText("jblow@work.com"));
        contactList.addContent(contact);

        contact = new Element("contact");
        contact.addContent(new Element("display")
            .setText("Doe, Jane"));
        contact.addContent(new Element("email")
            .setText("jane_doe@hotmail.com"));
        contactList.addContent(contact);

        contact = new Element("contact");
```

```
contact.addContent(new Element("display")
    .setText("Mickey Mouse"));
contact.addContent(new Element("email")
    .setText("mm@mousehouse.ca"));
contactList.addContent(contact);

contact = new Element("contact");
contact.addContent(new Element("display")
    .setText("Gordie Johnson"));
contact.addContent(new Element("email")
    .setText("gj@bigsugar.ca"));
contactList.addContent(contact);

document.getRootElement().addContent(contactList);
saveDocument(request, document);

//return a "success" since nothing could possibly go wrong
return mapping.findForward("success");
} //end perform
} //end TestSuiteAction
```

2. The XML Document

When the XML is passed to the transformation process, its contents will not only include the XML you've defined in your Action, but XML for all parameters and attributes in your request object, all errors in your ActionErrors object, the contents of your ApplicationResources file (included localized versions), all ActionMessages, and the ActionForm (if applicable).

```
<?xml version="1.0" encoding="UTF-8" ?>
<stxx>
  <contactListExample>
    <contactList>
      <contact>
        <display>Blow, Joe</display>
        <email>jblow@work.com</email>
      </contact>
      <contact>
        <display>Doe, Jane</display>
        <email>jane_doe@hotmail.com</email>
      </contact>
      <contact>
        <display>Mickey Mouse</display>
        <email>mm@mousehouse.ca</email>
      </contact>
      <contact>
        <display>Gordie Johnson</display>
        <email>gj@bigsugar.ca</email>
      </contact>
    </contactList>
  </form />
```

```
<messages />
<request>
  <param name="debug">
    <value>true</value>
  </param>
</request>
<applicationResources>
  <key name="run.example">Run the example</key>
  <key name="debug.example">View XML Source</key>
</applicationResources>
</contactListExample>
</stxx>
```

The request object will be populated with all the data being sent into the Action class along with all the request attributes that are being set in the Action class as well. The entire ApplicationResources file will be attached as well so that the transformation method has access to all of the defined key-value pairs defined in that properties file.