

Usage

by Don Brown

Advanced configuration

Table of contents

1 Overview.....	2
2 Implementing XMLForm.....	2
3 As a Struts ActionForm.....	2
4 As a SOAP Web Service.....	3

1. Overview

To create an XML form, you need to implement `com.oroada.stxx.xform.XMLForm`. If the form will be exposed as a Struts ActionForm, the easiest way is to extend `com.oroada.stxx.xform.DOMForm` or `com.oroada.stxx.xform.JDOMForm`. The XML Form can simultaneously be exposed as both a Struts ActionForm and a SOAP web service.

2. Implementing XMLForm

If you are simply wanting to create a Struts ActionForm that lets you modify XML, then you don't need to create your own implementation of `com.oroada.stxx.xform.XMLForm` or even extend `com.oroada.stxx.xform.DOMForm` or `com.oroada.stxx.xform.JDOMForm`, you can use one of the latter two classes directly. If you want to also expose the ActionForm as a web service, you will need to extend `com.oroada.stxx.xform.DOMForm` or `com.oroada.stxx.xform.JDOMForm` and override, at a minimum, the `save()` method. Also, you can override the `validate()` method to perform any extra validation your form requires. This is because when the XMLForm is exposed as a web service, it will not have the Struts Action to perform any saving operations, so this code has to be put in the XMLForm itself.

3. As a Struts ActionForm

Whether you use XMLForm.org's XForms implementation to generate an HTML form or not, the end result will be the same. The HTML form element names should be XPath statements pointing to the location in the XML that the values from the HTML input tags should change.

Note:

The node the XPath points to must exist in the XML model.

For example, for the XML form model:

```
<item id="foo">
  <name>Foo</name>
</item>
```

an HTML input text box used to change the name of the item would be defined as:

```
<input type="text" name="item/name" />
```

and an HTML hidden field for the item's id would look like:

```
<input type="hidden" name="item/@id" />
```

4. As a SOAP Web Service

To consume the SOAP web service, simply pass one or more XML nodes in a document-style SOAP call with no encoding in the schema expected by the XML form. Each XML node will be processed as an individual form. stxx will then use that XML node, validate it against the schema (if available), and call the XML form's save() method.

To create the return information, for each XML node that was sent, a result element is created. Inside the result element, either the string result of the save() method is placed or a list of errors from the failed validation.

For example, say the XML form was expecting something like this:

```
<item id="foo">
  <name>Foo</name>
</item>
```

and was sent this in the SOAP body:

```
<item id="bar">
  <name>Bar</name>
</item>
```

and upon validation, all checks passed so the save() method was called and returned "success". The XML stxx would return would be:

```
<result id="0">
  success
</result>
```

Now, if this XML was sent by the SOAP client:

```
<item>
  <name>Bar</name>
</item>
```

and the item's id attribute was required, the XML form's save() method would not be called and stxx would return something like this:

```
<result id="0">
  <error property="item/@id" name="item.id.missing">
    <text>Each item needs an id value</text>
  </error>
</result>
```