

How to build an XMLForm Wizard, Step Two

by Heidi Brannan

1. Step Two: Validation

The next step validation is optional, you do not have to validate your forms but its very easy to do so, so we will.

In the sitemap there are 2 parameters, `xmlform-validator-schema` and `xmlform-validator-schema-ns`. If these are left empty then no validation will be carried out. `xmlform-validator-schema` contains the name of the xml schema file we are using and `xmlform-validator-schema-ns` is the validator namespace we are using. Below are the 2 parameters in the sitemap we are referring to.

```
<map:parameter name="xmlform-validator-schema-ns"
  value="http://www.ascc.net/xml/schematron"/>
<map:parameter name="xmlform-validator-schema"
  value="schematron/wizard-xmlform-sch-report.xml"/>
```

Copy the "howto-xmlform-sch-report.xml" and place this in the folder `apache\xml-cocoon2\src\scratchpad\webapp\mount\xmlform\schematron`.

The schema is simple and only validates the registration.xml page. The phase tag relates to an xml page that contains values which will be validated. The id attribute contains the page name. Inside the phase tag is the active tag which contains the pattern attribute which match up to the pattern tags below. Each pattern can contain one or more rule tags. Each rule tag has a context attribute which maps to a javaBean value, for example `userName`. Nested inside the rule tag is one or more assert tag. Each assert tag contains a test attribute. The test value can check the value passes certain criteria, for example, the value is more than 7 characters long. Nested between the beginning and closing assert tags is an error message which will be

displayed if the test is not met.

```
<?xml version="1.0" ?>
<!--
  Validating Schematron schema for the xmlform example wizard

  Schematron Schema language home page:
  http://www.ascc.net/xml/schematron/

  Author: Ivelin Ivanov, ivelin@apache.org, April 2002
-->
<schema ns="http://xml.apache.cocoon/xmlform"
  xmlns="http://www.ascc.net/xml/schematron">

  <title>Schema for the XML Form example</title>

  <phase id="registration">
    <p>For user identity information.</p>
    <active pattern="reg"/>
  </phase>

  <phase id="confirm">
    <p>For final total validation and tracking
      some tricky problems.</p>
    <active pattern="reg" />
  </phase>

  <pattern name="User Info Validation Pattern" id="reg">
    <rule context="/userName">
      <assert test="string-length(.) > 7" diagnostics="dname dcount">
        Username should be at least 8 characters.
      </assert>
      <assert test="string-length(.) < 20">
        Username should be less than 20 characters.
      </assert>
    </rule>
    <rule context="/password">
      <assert test="string-length(.) > 7" diagnostics="dname dcount">
        Password should be at least 8 characters.
      </assert>
      <assert test="string-length(.) < 20">
        Password should be less than 20 characters.
      </assert>
    </rule>
    <rule context="/email">
      <assert test="contains( string(.),'@')">
        Email format is invalid.
      </assert>
    </rule>
  </pattern>
</schema>
```

How to build an XMLForm Wizard, Step Two

```
</assert>  
</rule>  
</pattern>  
</schema>
```

Now you are ready for [Step 3: Java Bean](#)

2. Revisions

Find a problem with this document? Consider contacting the mailing lists or submitting your own revision. For instructions, read the [How To Submit a Revision](#).