1 **From:** simon_g_brown <simon_g_brown@y...>
Date: Tue Apr 30, 2002 6:10am
Subject: Background

There are currently several different ways in which JSP custom tags can be tested. One of these is to mock out the classes that make up the JSP API such as PageContext, ServletRequest and so on. A downside of this approach when testing tags is that you are moving away from the actual environment in which the tags are running. In addition to this, it makes unit testing on different containers much harder. Another way to test tags is to simply include them in the JSPs, and test the JSPs with something like HttpUnit. This is a good approach but does make it harder to test the tags in isolation.

Custom tags are reusable components that encapsulate logic for use for page authors and as such, should be tested independently of the pages on which they are used. Ideally, tags should represent actions that happen on the page, although it is often appropriate to have them generate small amounts of dynamic content too. In summary, here is a list of the functions that a tag may perform:

- Generating dynamic content
- Introducing scripting variables
- Programmatic evaluation of body content
- Multiple evaluations of their body content

From a testing perspective, tags may require the following types of testing:

- Correctness of generated content
- Correctness of introduced scripting variables
- Correctness of programmatic evaluation of body content
- Correctness of multiple evaluations of their body content
- Correctness of tag usage with/without optional attributes

In addition to "unit testing" tags, it is also useful to be able to test the cooperation between tags in a tag library, to ensure that the correct results are produced when tags are used in conjunction with one another. This is more like component testing than unit testing, but is still important and possible within the same framework.

The purpose of this Yahoo! Group is to throw some ideas around for how a framework for testing tags can be developed, and to subsequently build the framework.

2 **From:** simon_g_brown <simon_g_brown@y...>
Date: Tue Apr 30, 2002 6:30am
Subject: First thoughts

Here are my first thoughts on how we could test tags from inside the container. Essentially, we just wrap up JSP content (tags, scriptlets, expressions, etc) inside some specific assertion tags. Behind the scenes, these assertion tags simply consume the body content and when the end tag is reached, they assert the condition.

```
<html>
```

```
  <head>
    <title>TagUnit</title>
    <link rel="stylesheet" href="./page.css">
  </head>
  <%@ taglib uri="/tagUnit" prefix="test" %>
  <body>

    <h1>TagUnit example test suite</h1>

    <test:testSuite name="Simple Test Suite">

      <test:testCase name="Simple tests">

        <test:assertEquals name="1=1">
          <test:expectedResult>1</test:expectedResult>
          <test:actualResult>1</test:actualResult>
        </test:assertEquals>

        <test:assertEquals name="Scriptlet test">
          <test:expectedResult><%= "hello" %></test:expectedResult>
          <test:actualResult>hello</test:actualResult>
        </test:assertEquals>

      </test:testCase>

      <test:testCase name="Random tests">

        <test:assertEquals name="Random number test">
          <test:expectedResult><%= Math.random() %>
</test:expectedResult>
          <test:actualResult><%= Math.random()%>
</test:actualResult>
        </test:assertEquals>

      </test:testCase>

    </test:testSuite>

    <!-- this is simply for presentation -->

    <table>
      <test:testCaseResults>
        <tr>
          <td>
            <jsp:getProperty name="testCaseResult" property="name"/>
          </td>
          <td>
            <jsp:getProperty name="testCaseResult"
property="success"/>
          </td>
        </tr>
      </test:testCaseResults>
    </table>

  </body>
```

```
</html>
```

A benefit of testing inside the container is that you have full access to the infrastructure - JSP/Servlet/other tags, EJBs, JDBC, etc. Also, you now have a way to properly test on different containers - just redeploy and you're off. This would be great to help catch tag pooling/reuse problems on containers that support these optional features.

At the moment, the tags just assert based upon generated content. What do people think of this? We could also have tags that assert based upon scripting variables/page context attributes. Generated content, however, does provide an easy way of testing introduced scripting variables too since we can simply output them to the JSP and test them against a regular scriptlet (for example). Whitespace (spaces, tabs, etc) could be a problem (as they often are in JSP/XML in general) but we could have the ability to ignore it.

Some of my other thoughts were that we could have individual JSPs as test cases, and have them included (through a dynamic, server-side include) into a test suite. Something of the form...

```
<testSuite name="x">
  <testCase uri="abc.jsp/>
</testSuite>
```

At the moment, all result state is being maintained internally, but my thoughts are around having one or more request scoped, well-known named attributes so that we split tests over multiple JSPs.

As for presenting the results, I am thinking that something like HttpUnit or XMLUnit could be used to assert that the tests were successful. We could output the results as an XML document, or just a very simple HTML document. This means that the tests can be automated (e.g. from ant) once the app has been deployed. For reporting and visual testing, people could write a nicely formatted page showing the results. We could even write some tags that help with this task.

I think that there are three parts to this then:

(1) the testing and assertion tags
(2) the interface to HttpUnit/XMLUnit
(3) the tags to help with showing the results

So then ... thoughts, comments, etc? Is this to far off the mark? Are there any scenarios that can't be tested easily? Tags that support multiple browsers are testable, since we can fake the browser type from HttpUnit (that's right isn't it?).

3 **From:** simon_g_brown <simon_g_brown@y...>
Date: Tue Apr 30, 2002 7:17am
Subject: JUnit for the in-container tests?

Something that I was thinking about is whether to use JUnit for the in-container part of the framework - i.e. the custom tags. What do you think?

I have a feeling that it's not going to offer much benefit, simply because it is geared towards running testXXX() methods. The tags approach mimics the JUnit concepts, but doesn't use them under the covers. I think that the client-side of the framework should though.

Any thoughts?

5 **From:** simon_g_brown <simon_g_brown@y...>
Date: Tue Apr 30, 2002 7:23am
Subject: Automatic location of test cases

Coming back to the point about having a JSP = a test case, it's fairly straightforward to do the following:

```
<testSuite name="X">
  <testCases uri="xTagTests"/>
</testSuite>
```

Here, the testCases tag could look for all JSPs underneath the xTagTests directory and run them all.

Any thoughts?

6 **From:** chanochwiggers <chanoch@k...>
Date: Wed May 1, 2002 5:04am
Subject: Re: First thoughts

I like your Ideas, Si. It was time that someone joined though so you are not talking to yourself too much :)
With regards to the whitespace, I think there might be a need for a special character in the output that shows areas where whitespace can be ignored - that means that in areas where whitespace is important you can include it and it will be tested for, while in other areas you can specify that it be ignored.

I wonder whether its worth prototyping some stuff so that people can have a go and garner some opinions through experience.

Chanoch

7 **From:** Sam Dalton <sam.dalton@c...>
Date: Wed May 1, 2002 5:03am
Subject: RE: Re: First thoughts

I definitely agree that prototying may generate some discussion. I think that putting together a SIMPLE prototype would be good, then we can look at where existing efforts fit in (such as JUnit, Cactus, HttpUnit, XMLUnit etc), as well as seeing what other features would be useful.

S

8 **From:** Simon Brown <simon_g_brown@y...>
Date: Wed May 1, 2002 5:15am
Subject: Re: Re: First thoughts

Prototype ... yes I've been having a play around with ideas so I'll bundle them up and stick on the group site for people to have a play with.

I had some more ideas about the sort of things that you might want to test and I started putting together a test suite for a simple iteration tag (sorry if the spaces get lost in the e-mail!)

```
  <tagunit:testSuite name="Iteration taglib tests">

    <tagunit:testCase name="Iteration of body
content">

      <%
        Collection coll = new ArrayList();
        coll.add("1");
        coll.add("2");
        coll.add("3");
      %>

      <tagunit:assertEquals name="Iteration test -
String">

<tagunit:expectedResult>123</tagunit:expectedResult>
        <tagunit:actualResult><iteration:iterate
id="element" type="java.lang.String" collection="<%=
coll %>"><%= element
%></iteration:iterate></tagunit:actualResult>
      </tagunit:assertEquals>

      <%
        coll = new ArrayList();
        coll.add(new Integer(1));
        coll.add(new Integer(2));
        coll.add(new Integer(3));
      %>

      <tagunit:assertEquals name="Iteration test -
Integer">

<tagunit:expectedResult>123</tagunit:expectedResult>
        <tagunit:actualResult><iteration:iterate
id="element" type="java.lang.Integer" collection="<%=
coll %>"><%= element
%></iteration:iterate></tagunit:actualResult>
      </tagunit:assertEquals>

    </tagunit:testCase>

    <tagunit:testCase name="Introduction of scripting
variables">

      <%
        List list = new ArrayList();
        list.add(new Integer(1));
        list.add(new Integer(2));
        list.add(new Integer(3));
    int i = 0;
   %>

   <iteration:iterate id="element"
type="java.lang.Integer" collection="<%= list %>">
    <tagunit:assertScriptingVariable
name="element" type="java.lang.Integer" value="<%=
list.get(i) %>"/>
    <% i++; %>
   </iteration:iterate>

  </tagunit:testCase>

 </tagunit:testSuite>
```

Taking this as an example, what else might you want to
test at this level? (i.e. a unit, in this case a tag,
in isolation)

9 **From:** chanochwiggers <chanoch@k...>
Date: Wed May 1, 2002 5:16am
Subject: Re: First thoughts

I think that in terms of testing for side effects, you are
absolutely
going to have to test variables.

Also, is there a need for testing for the existence of cookies?

10 **From:** tagunit@y...
Date: Wed May 1, 2002 5:36am
Subject: New file uploaded to tagunit

Hello,

This email message is a notification to let you know that
a file has been uploaded to the Files area of the tagunit
group.

```
  File        : /prototypes/tagunit.war
  Uploaded by : simon_g_brown <simon_g_brown@y...>
  Description : A first prototype of some of the concepts
```

You can access this file at the URL

http://groups.yahoo.com/group/tagunit/files/prototypes/tagunit.war

To learn more about file sharing for your group, please visit

http://help.yahoo.com/help/us/groups/files

Regards,

simon_g_brown

11 **From:** simon_g_brown <simon_g_brown@y...>

Date: Wed May 1, 2002 6:06am
Subject: Exceptions

In JUnit, any exceptions are propagated up to the top and caught by
the test case rather than abnormally stopping the test. Since JSP
1.2
tags can implement TryCatchFinally, there's nothing stopping us from
doing the same. This is a nice way of catching exceptions that might
get thrown within the tag handler, the body content, if a scripting
variable hasn't been declared and so on.

In fact, we could also have asserts on tag usage and attribute
validation - catching exceptions that are thrown in the tag handler
during the tag invocation.

Hmmm...

12 **From:** Kunnumpurath, Meeraj <meeraj.kunnumpurath@e...>
Date: Wed May 1, 2002 6:21am
Subject: RE: Exceptions

Aren't validations on tag usage and attributes provided by the tag
API itself?

13 **From:** Simon Brown <simon_g_brown@y...>
Date: Wed May 1, 2002 6:28am
Subject: RE: Exceptions

They are, but I was just thinking that you might want
to test that the correct exceptions get thrown when
tags are used incorrectly.

For example, you may validate an attribute at
request-time and throw a JspTagException back to the
page signalling that something is wrong. As a part of
your test suite, you might want to test this.

What do you think ... useful/overkill/etc?

14 **From:** Kunnumpurath, Meeraj <meeraj.kunnumpurath@e...>
Date: Wed May 1, 2002 6:34am
Subject: RE: Exceptions

I agree .....:-)

15 **From:** Simon Brown <simon_g_brown@y...>
Date: Wed May 1, 2002 6:38am
Subject: RE: Exceptions

Useful AND overkill? ;-)

I guess that you'd just need to catch the exception,
check that it's of the correct type, and then check
the message via the getMessage() method.

16 **From:** simon_g_brown <simon_g_brown@y...>
Date: Thu May 2, 2002 4:37am

Subject: A more advanced prototype

I had a play with some of the ideas last night, particularly around handling exceptions, testing scripting variables, attributes and including other JSPs containing test cases. Actually, this has a nice side-effect in that in addition to making the test cases more structured, you can catch JSP compilation errors that are a result of (for example) a mismatch in the way that the tag works and is supposed to work through the TLD. For example, an incorrect scripting variable, incorrect body content type. I've put some logic in to catch these exceptions and keep a note of them.

I've also cleaned some stuff up a little, rearranged some bits and also, did some quick and dirty work on making a nice page on which to display the results.

There's a new WAR file in the "Files" section so if you get a chance, I'd be interested to hear what you think... :-)

From: chanochwiggers <chanoch@k...>
17 **Date:** Wed May 1, 2002 4:33pm
Subject: Re: A more advanced prototype

This looks pretty good Si, having a play at the moment. JRun doesnt have a problem with much of it. There's one issue where an exception is caught giving the error:

```
Translator.CompilationFailedExceptionCompiler errors:
Found 1 semantic error compiling "C:/JRun4/servers/default/SERVER-INF/temp/tagunit.war/WEB-INF/jsp/jrun__scripting2dvariables2dfail12ejsp1e.java":

    84. _tag2.setVariable( MyElement );
                           <-------->
*** Error: No entity named "MyElement" was found in this environment.
```

It comes from the lines:

```
    <tagunit:assertScriptingVariable name="Scripting variable" variable="<%= MyElement %>" type="java.lang.Integer" value="<%= list.get(i) %>"/>
    <tagunit:assertPageContextAttribute name="Page context attribute" id="element" type="java.lang.Integer" value="<%= list.get(i) %>"/>
```

which should read:

```
    <tagunit:assertScriptingVariable name="Scripting variable" variable="<%= element %>" type="java.lang.Integer" value="<%= list.get(i) %>"/>
    <tagunit:assertPageContextAttribute name="Page context attribute" id="MyElement" type="java.lang.Integer" value="<%= list.get(i) %>"/>
```

Once you do that it all works fine.

18 **From:** chanochwiggers <chanoch@k...>

Date: Wed May 1, 2002 4:35pm
Subject: http://www.simongbrown.com/tagunit/

How about testing for the existence and values of cookies?
Do you think its worth supporting?

Chanoch

From: chanochwiggers <chanoch@k...>
19 **Date:** Wed May 1, 2002 4:42pm
Subject: SetUp and TearDown

Have you used SetUpTag and TearDownTag tag?
What kind of resources do you think you might have to setup and tear down here?
Database related resources?

From: Kunnumpurath, Meeraj <meeraj.kunnumpurath@e...>
20 **Date:** Wed May 1, 2002 4:56pm
Subject: RE: SetUp and TearDown

I think setup would be a nice tag to nest the tag you want to test.
For eg.

```
<tagunit:setup>
        <%-- This tag will introduce a scripting variable -->
        <mine:myTag id="xyz" value="10"/>
</tagunit:setup>

<tagnuint:assertScriptingVatiable name="xyz" value="10"/>
```

From: Simon Brown <simon_g_brown@y...>
21 **Date:** Wed May 1, 2002 5:38pm
Subject: Re: Re: A more advanced prototype

That's right ... I was just showing that we can also
catch page compilation errors and show them to the
"user" - whether that be a proper user or an automated
user. Tomcat even includes the correct tabulation in the
error messages so that you can display it with <pre>
tags! :-)

From: Simon Brown <simon_g_brown@y...>
22 **Date:** Wed May 1, 2002 5:40pm
Subject: Re: Cookies

If tags are able to create cookies, which they can,
then it's certainly worth putting in. Now, I wonder
where I have seen a tag that checks for a cookie...
;-)

Something like <assertCookie/>, with name and an
optional value attribute perhaps?

From: Simon Brown <simon_g_brown@y...>
23 **Date:** Wed May 1, 2002 5:43pm

Subject: RE: SetUp and TearDown

Well this was the idea that I first had since it
mimics the JUnit way of setting the tests. However,
there is a small flaw that I had not thought about -
if you are initialising/introducing scripting
variables from inside the setUp tag, they are not
visible outside of the setUp tag since the compiled
servlet sticks and try-catch block around it all.

As you'll see from the body-content.jsp page in the
current bundle, you can put code before the asserts.
This achieves the same thing but isn't as explicit.

Any thoughts?

From: Kunnumpurath, Meeraj <meeraj.kunnumpurath@e...>
24 **Date:** Wed May 1, 2002 6:25pm
Subject: RE: Cookies

There is a Struts custom tag that checks for a cookie..

From: Simon Brown <simon_g_brown@y...>
25 **Date:** Wed May 1, 2002 6:51pm
Subject: RE: Cookies

Sorry, I was fooling around - there are some cookie
tags in my book. :))

From: simon_g_brown <simon_g_brown@y...>
26 **Date:** Thu May 2, 2002 6:44pm
Subject: TLD constraints and test structure

Been doing some thinking of what else you might want to test and
came up with a couple of ideas about testing the TLD constraints.
For example, asserting that a particular attribute is mandatory,
body content is empty, etc.

I thought of doing something like this but it's not very elegant:

```
  <tagunit:assertAttribute name="Mandatory href attribute
specified">
    <html:a href="/simple">Simple link</html:a>
  </tagunit:assertEquals>

  <tagunit:assertAttribute name="Mandatory href attribute not
specified">
    <html:a>Simple link</html:a>
  </tagunit:assertEquals>
```

Instead, what about something like this:

```
  <tagunit:assertAttribute name="href" required="true"
rtexprvalue="false"/>
```

Since we can't (unfortunately) get to the TLD details via the
container, we could define the location (as a URI) when defining the
test suite.

My current thinking is that you would define a test suite for a particular tag library. What do you think?

In my current prototype (which I will upload later or at the weekend), I have the following directory structure :

```
/
| - html
|    | index.jsp
|    |- base
|    |   |
|    |   test-XXX.jsp
|    |
|    |- anchor
|        |
|        test-<XXX>.jsp
|
| - <taglib name>
|    | index.jsp
|    |- <tagname>
|    |   |
|    |   test-<XXX>.jsp
```

So the index.jsp under html is the start of the test suite for the html tag library, and all test cases for the anchor tag are under the anchor directory, and so on. Again, any thoughts?

From: simon_g_brown <simon_g_brown@y...>
27 **Date:** Mon May 6, 2002 6:46pm
Subject: Re: TLD constraints and test structure

I've been having a play and have renamed some of the tags to reflect the structure outlined in the previous posting as follows:

```
  <tagunit:testTagLibrary name="HTML tag library">

    <tagunit:tagLibraryDescriptor jar="/WEB-INF/lib/html.jar"
name="html.tld"/>

    <tagunit:testTag name="a" uri="/html/a"/>
    <tagunit:testTag name="base" uri="/html/base"/>

  </tagunit:testTagLibrary>
```

I have also created the tagLibraryDescriptor tag from which you can specify the TLD. This tag simply parses the file and creates a whole bunch of objects behind the scenes (similiar to the way that the container does it). From here, it is easy to run asserts over the contents - for example:

```
<tagunit:testCase name="Attributes">

  <tagunit:assertAttribute name="href" required="true"/>
  <tagunit:assertAttribute name="target" required="false"/>
  <tagunit:assertAttribute name="secure" required="false"/>
```

```
</tagunit:testCase>
```

We could also add rtexprvalue, etc too. In fact, the assertAttribute
tag could load up the tag handler class and check for the presences
of an appropriate setter method.

This would be useful to catch those occassions where you change the
required attribute and don't test the existing pages - which I have
done before. ;-)

Any thoughts?
Simon

28 **From:** Chanoch Wiggers <Chanoch.Wiggers@k...>
Date: Tue May 7, 2002 2:00pm
Subject: RE: Re: TLD constraints and test structure

Is it worth having tests for what interfaces a tag implements? If
you
have a class hierarchy for a family of tags, you may in the future
decide that you have to implement a different helper class/tag
interface/own interface. I can think of times when it would be worth
knowing a) if there is content in the tag and it implements a body
less
interface, or b) if it should extend/implement a specific
class/interface

chanoch

29 **From:** simon_g_brown <simon_g_brown@y...>
Date: Tue May 7, 2002 2:17pm
Subject: Re: TLD constraints and test structure

Sounds like a good idea to me - and it's very easy to do since the
<tagLibraryDescriptor> tag currently reads in/parses the TLD file.

Any thoughts on how you might present this, something like the
following perhaps? (bearing in mind that the taglib and tag context
will already have been setup)

```
<assertInterface type="com.abc.MyTagInterface"/>
```

Simon

30 **From:** simon_g_brown <simon_g_brown@y...>
Date: Tue May 7, 2002 2:21pm
Subject: www.tagunit.org...

...is up, although in skeleton form. :-)

31 **From:** simon_g_brown <simon_g_brown@y...>
Date: Wed May 8, 2002 7:38pm
Subject: TagUnit @ SourceForge

Right ... the project is now registered with SourceForge so I'll
start moving some bits and peices over in the next few days once
I've

figured how to work everything works.

The project page can be found at http://sourceforge.net/projects/tagunit/ and http://tagunit.sourceforge.net will be available (via DNS anyway) shortly so I'll move the project website over too.

Cheers
Simon

From: ersin_eser <ersin_eser@y...>
32 **Date:** Thu May 9, 2002 5:45pm
Subject: Re: TagUnit @ SourceForge

Cool, I have just collected your posts here into word doc to read them all at once.
See you over sourceforge
regards,
ersin

From: simon_g_brown <simon_g_brown@y...>
33 **Date:** Fri May 10, 2002 5:24am
Subject: Re: TagUnit @ SourceForge

Hmmm ... that sounds like a good idea to me. :-) Since sourceforge provides mailing lists and forums, I think that we'll move over.

Any thoughts as to whether we should this one going to? (I can delete
it if we want)

Anybody know if there is an easy way to "export" all yahoo group messages to a file? Or perhaps, Ersin, any chance that you could e-mail me over your Word doc and I'll try to upload it to sourceforge somewhere.

Cheers
Simon

From: simon_g_brown <simon_g_brown@y...>
34 **Date:** Fri May 10, 2002 6:00am
Subject: tagunit-devel mailing list

I've setup a bunch of mailing lists and am in the process of configuring them up - there's tagunit-devel, tagunit-user and tagunit-announce.

When you get a minute, please sign up to the new one at https://lists.sourceforge.net/lists/listinfo/tagunit-devel :-)

Thanks
Simon