HTTPUNIT

Lauri Peterson
Technology Partners, Inc
Lauri.peterson@monsanto.com

Topics Covered

- What is Httpunit
- WebConversation
- Simple example
- Working with forms
- Working with tables
- Web navigation
- Javascript
- When output is XML
- Demos
- ServletUnit
- Not Covered
- Summary

What Is Httpunit?

- Way to dynamically test web pages
  - Does a link take me to the right page?
  - Did the page render correctly?
  - Did Javascript enable the button?
- Java API for web interaction
  - WebWindow, WebForm, WebLink
- Browser emulation
  - Don’t need a browser running
  - Do need a web server (i.e. Apache/Tomcat)
  - Some support for Javascript

What Is Httpunit (cont)?

- Free!
  - Not perfect, but it is useful
  - Inconsistently named classes & methods
  - Javascript not fully supported
- Compliments JUnit tests
  - Black box testing
  - Automation of testing user interaction with web pages
  - Business logic should still be tested elsewhere
- ServletUnit test Servlets without container
  - Mocks out the servlet container

WebConversation

- Emulates the request/response

```
WebConversation converstaion = new WebConversation();
WebRequest request = new PostMethodWebRequest(“URL”);
WebResponse webResponse = conversation.getResponse(request);
```

- If web authorization is needed

```
conversation.setAuthorization(“userName”, “password”);
```

As when this is used in Apache:

```
<Location /httpUnit>
      AuthType Basic
      AuthName “Please log in"
      AuthUserFile dir/user.pass
      require valid-user
</Location>
```

Simple Example

```
WebConversation converstaion = new WebConversation();
WebRequest request =  new PostMethodWebRequest("http://localhost/httpunit/html/page1.html");
webResponse = conversation.getResponse(request);

assertEquals("HTTPUnit Example: Intro Page 1", webResponse.getTitle());

//get raw HTML for the page so we can check for some basic things we are expecting
String pageText = webResponse.getText();

assertTrue(pageText.indexOf("Intro Page 1") > -1);
```

Tests see if text is on page or not

```
// also want to make sure some special formatting was there
assertTrue(pageText.indexOf("<H2 align=\"center\">HTTPUnit Example: Intro Page 1</H2>") > -1);

// may want to test that some thing (like an error message?) does NOT appear on the screen
assertTrue(pageText.indexOf("Fatal error occurred:") == -1);

// if order of information on the screen was important
int locationOfString1 = pageText.indexOf("Intro Page 1");
int locationOfString2 = pageText.indexOf("Different Kinds of Form Inputs");
assertTrue((locationOfString1 > -1) && (locationOfString1 < locationOfString2));
```

Forms – 3 ways to find it

```
 WebForm formFromArray = webResponse.getForms()[0];
assertNotNull(formFromArray);

// or modify your html to add id= to the form tag
//<FORM METHOD="POST“…name="FormName" id="FormId">
WebForm formFromId = webResponse.getFormWithID(FORM_ID);
assertNotNull(formFromId);

// or modify your html to add name= to the form tag
//<FORM METHOD="POST“… name="FormName" id="FormId">
WebForm formFromName = webResponse.getFormWithName("FormName");
assertNotNull(formFromName);
```

If you know the order the forms appear on the page

By name or id where possible

Forms - Text

- Looks just like HttpRequest

```
// <INPUT TYPE="TEXT" NAME="textInput" VALUE="" SIZE="45" MAXLENGTH="45">
  assertEquals("", webForm.getParameterValue(“textInput”));
  assertTrue(webForm.isTextParameter(“textInput”));
  assertFalse(webForm.isDisabledParameter(“textInput”));
  assertFalse(webForm.isReadOnlyParameter(“textInput”));
```

- Values can be modified too

```
  webForm.setParameter(“textInput”, "newValue");
  assertEquals("newValue", webForm.getParameterValue(“textInput”));
```

Forms – Text (cont)

- Disabled and Readonly too

```
//<INPUT TYPE="TEXT" NAME="textInputDisabled" VALUE="this is disabled" disabled="true">
assertEquals("this is disabled", webForm.getParameterValue("textInputDisabled"));
assertTrue(webForm.isTextParameter("textInputDisabled"));
assertTrue(webForm.isDisabledParameter("textInputDisabled"));
assertTrue(webForm.isReadOnlyParameter("textInputDisabled"));  //???
```

Why?

```
webForm.setParameter("textInputDisabled", "newValue");
```

Wrong!

```
//<INPUT TYPE="TEXT" NAME="textInputReadonly" VALUE="this is readonly" readonly="true">
assertEquals("this is readonly", webForm.getParameterValue("textInputReadonly"));
assertTrue(webForm.isTextParameter("textInputReadonly"));
assertFalse(webForm.isDisabledParameter("textInputReadonly"));
assertTrue(webForm.isReadOnlyParameter("textInputReadonly"));

webForm.setParameter("textInputReadonly", "newValue");
```

Wrong!

Forms – password & hidden

- Just like a text field

```
//<INPUT TYPE="PASSWORD" NAME="passwordInput" VALUE="" SIZE="45" MAXLENGTH="45">
  assertEquals("", webForm.getParameterValue("passwordInput"));
  assertTrue(webForm.isTextParameter("passwordInput"));
  webForm.setParameter("passwordInput", "newValue");
  assertEquals("newValue", webForm.getParameterValue("passwordInput"));

//<INPUT TYPE="HIDDEN" NAME="hiddenInput" VALUE="hiddenValue">
  assertEquals("hiddenValue", webForm.getParameterValue("hiddenInput"));
  assertTrue(webForm.isHiddenParameter("hiddenInput"));
  assertFalse(webForm.isReadOnlyParameter("hiddenInput")); //???
```

Why?

```
  webForm.setParameter(HIDDEN_INPUT, "newValue");
```

Wrong!

Forms - checkboxes

- Has own setter method
- using setParameter throws an exception

```
//<input type="checkbox" name="checkBoxInput1" value="true" checked="true">check box 1</input>
//<input type="checkbox" name="checkBoxInput2" value="true" checked="true">check box 2</input>
//<input type="checkbox" name="checkBoxInput3" value="true" checked="true">check box 3</input>
  assertEquals("true", webForm.getParameterValue("checkBoxInput1"));

  webForm.setParameter("checkBoxInput1", "false");
```

Wrong!

```
  webForm.setCheckbox("checkBoxInput1", false);
  assertEquals(null, webForm.getParameterValue("checkBoxInput1"));
```

Forms – checkboxes (cont)

- When names are the same

```
//<input type="checkbox" name="aCheckbox" value="checkBoxA" checked="true"> 1</input>
//<input type="checkbox" name="aCheckbox" value="checkBoxB" checked="true"> 2</input>
//<input type="checkbox" name="aCheckbox" value="checkBoxC" checked="true"> 3</input>

 assertEquals("checkBoxA", webForm.getParameterValue("aCheckbox"));
```

Works, but just
Get first value??

```
 assertEquals(3, webForm.getParameterValues("aCheckbox").length);
 webForm.setCheckbox("aCheckbox", "checkBoxB", false);
 String[] nowChecked = webForm.getParameterValues("aCheckbox");
 assertEquals("checkBoxA", nowChecked[0]);
 assertEquals("checkBoxC", nowChecked[1]);
```

Extra param

```
 webForm.setCheckbox("aCheckbox", false);
```

Throws exception

Forms – radio buttons

- No special setter for radio buttons

```
//<input type="radio" name="radioInput" value="radio1" checked="true">radio button 1</input>
//<input type="radio" name="radioInput" value="radio2">radio button 2</input>

  assertEquals("radio1", webForm.getParameterValue("radioInput"));
  webForm.setParameter("radioInput", "radio2");
  assertEquals("radio2", webForm.getParameterValue("radioInput"));

  webForm.setParameter("radioInput", "junk");
  String[] tryToSetBoth = {"radio1", "radio2"};
  webForm.setParameter("radioInput", tryToSetBoth);

  webForm.setCheckbox("radioInput", "radio1", true);
```

Throws exceptions

Forms – Select

```
//<SELECT NAME="selectSingle" SIZE="1" >
//    <OPTION  VALUE="" SELECTED/>
//    <OPTION VALUE="opt1">Option Number 1</option>
//    <OPTION VALUE="opt2">Option Number 2</option>
//    <OPTION VALUE="opt3">Option Number 3</option>
//  </SELECT>
```

Didn’t like this:
<OPTION SELECTED VALUE="" />

```
assertEquals("", webForm.getParameterValue("selectSingle"));
assertFalse(webForm.isMultiValuedParameter("selectSingle"));
webForm.setParameter("selectSingle", "opt2");
assertEquals("opt2", webForm.getParameterValue("selectSingle"));

webForm.setParameter(("selectSingle", "junk");
webForm.setParameter("selectSingle", "");
```

Wrong!

```
String[] tryToSetTwo = {"opt1", "opt2"};
webForm.setParameter(("selectSingle", tryToSetTwo);
```

Wrong!

Forms – Select (cont)

```
//<SELECT NAME="selectMultiple" SIZE="3"  multiple="TRUE">
  //    <OPTION  VALUE="" SELECTED></option>
  //    <OPTION VALUE="opt1">Option Number 1</option>
  //    <OPTION VALUE="opt2">Option Number 2</option>
  //    <OPTION VALUE="opt3">Option Number 3</option></SELECT>
```

Didn't like this:
<OPTION VALUE="" SELECTED/>

```
assertEquals("", webForm.getParameterValue("selectMultiple"));
assertEquals(1, webForm.getParameterValues("selectMultiple").length);
assertTrue(webForm.isMultiValuedParameter("selectMultiple"));

webForm.setParameter("selectMultiple", "opt2");
```

Changes, does not add to selection

```
String[] currentValues = webForm.getParameterValues("selectMultiple");
assertEquals(1, currentValues.length);
assertEquals("opt2", currentValues[0]);

webForm.setParameter("selectMultiple", {"opt1", "opt2"});
```

Select all at once

```
currentValues = webForm.getParameterValues("selectMultiple");
assertEquals(2, currentValues.length);
```

Forms - TextArea

- Treats as a single string

```
//<TEXTAREA NAME="textArea" VALUE="" rows="3" cols="45"></TEXTAREA>
   assertEquals("", webForm.getParameterValue("textArea"));
   assertTrue(webForm.isTextParameter("textArea"));
   webForm.setParameter("textArea", "newValue\nline 2");
   assertEquals("newValue\nline 2", webForm.getParameterValue("textArea"));
```

Forms - buttons

- Many ways to find the button

Why Button instead of WebButton?

Why no getButtonWithName?

```
Button buttonById = webForm.getButtonWithID("resetButtonId");
```

```
Button[] allButtons = webForm.getButtons();
```

In the order they appear on the form,
But that is not stated in the documentation

Special forms for submit buttons

Shouldn’t this be
getSubmitButtonWithName?

```
Button buttonByName = webForm.getSubmitButton("submitButtonName");
```

```
Button buttonById = webForm.getSubmitButtonWithID("submitButtonId");
```

Forms - submit

- Button click – must refresh WebResponse

```
Button buttonByName = webForm.getSubmitButton("submitButtonName");
buttonByName.click();
webResponse = conversation.getCurrentPage();
```

Must get a handle to the New WebResponse

- Form submit – returns WebResponse

```
webResponse = webForm.submit();

webForm.submit(buttonByName); // if there is more than 1 submit button on the form
```

Tables

- Multiple ways to find a table

```
WebTable[] allTables = webResponse.getTables();
assertEquals(2, allTables.length);
WebTable tableData1 = allTables[0];
```

In the order they appear on the page

```
WebTable tableData4 = webResponse.getTableWithID("tableId");
```

By value of the first cell

```
WebTable tableData2 = webResponse.getTableStartingWith("first table, first row, column 1");
```

By value of the first part of the first cell

```
WebTable tableData3 = webResponse.getTableStartingWithPrefix("first table");
```

Why no getTableWithName?

Tables - cells

- Better way to test data order

Index is zero based

```
String value1 = tableData.getCellAsText(2,2));
String value2 = tableData.getCellAsText(3,2));
```

- Can also check table format

```
TableCell cell = tableData.getTableCell(2,2);
assertEquals(1, cell.getRowSpan());
assertEquals(1, cell.getColSpan());
```

Navigation - links

- Find the link by the text

```
WebLink link = webResponse.getLinkWith(“text that user sees on the screen");
webResponse = link.click();
```

Button.click did not return the webResponse
But link does…

Navigation (cont)

- When the new page opens in a new browser window

```
//<FORM METHOD="POST" ACTION="lastPage.html"  target="window2">
    buttonByName.click();

   webResponse = conversation.getCurrentPage();
```

the new response opened in a new window, but doing this just gives the original window

```
   WebWindow[] windows = conversation.getOpenWindows();
   assertEquals(2, windows.length);
   webResponse = windows[1].getCurrentPage();
```

Need to do this

```
   webResponse = conversation.getOpenWindow("window2").getCurrentPage();
```

or this

```
   webResponse = webForm.submit();
```

In this case this returns the new window

Javascript

- Support for window.alert

```
function validateForm() {
  window.alert("this is a javascript alert!");
  return false;
}

assertEquals("this is a javascript alert!", conversation.popNextAlert());
```

Returns “” (empty string) when there are no alerts

Javascript

- Support for window.open

```
function openWindow(url) {
     window.open(url, "dialog", "width=200,height=150,status=no,resizable=no,top=200,left=200");
}
 <a href="javascript:openWindow('simpleDialog.html');">simple dialog </a>

   WebLink link = webResponse.getLinkWith("simple dialog");
   webResponse = link.click();

   webResponse = getNewlyOpenedBrowserWindow();  // my convenience method
   WebForm webForm = webResponse.getFormWithID(“formId);
   webForm.getButtonWithID("close me").click();

   WebWindow[] windows = conversation.getOpenWindows();
   assertEquals(1, windows.length);
```

In this case it does NOT return the new window

XML responses

- Will return the XML in DOM format

```
Document dom = webResponse.getDOM();
NodeList nodeList = dom.getElementsByTagName("item1");
…
```

Other Features

- Support for cookies
- Support for frames

Other tidbits

- Forget to start webserver
- Changes to test do not require redeploy
- Changes to source code DO
- Forgetting conversation.getCurrentPage
- Compression bug

```
// get: java.io.EOFException: Unexpected end of ZLIB input stream
conversation.getClientProperties().setAcceptGzip(false);
```

For example in apache:

```
<Location />
  SetOutputFilter DEFLATE
</Location>
```

More Info

- Httpunit.sourceforge.net
- Cookbook
- FAQ
- Javadocs