

GT2004Nico

Test driven development with Cocoon

Nico Verwer works for Be Value, for a large publisher in legal and text publications.

Client had many problems with their content

Cocoon enabled them to make a prototype within one month. They felt that cocoon was lacking a testing framework.

How to use testing to find the bugs, how to use testing to help you write efficient code.

Test driven development

1. Write a simple test

2. make sure the test fails (you have not written the code) 3. write only the minimum code for the test to pass 4. make sure the test passes 5. refactor the code to make sure it has the simplest design to do the job

Not new, very iterative way for development. See JUnit website. How to use this for cocoon? Developers have a tendency to write really nice code that does more than intended.

Unit and Regression testing

re-run the tests over and over, to detect when a change breaks a test Only check in the code when all of the tests succeed.

A bug is found (maybe you forgot to add a test). Find the bug, add to the tests, fix the bug, run the tests.

Organize your tests in suites.

Testing frameworks

Developers don't like to write tests. Project managers don't like to write tests. So it should be made easy. Testing appears to take a bit of time, but in the long run it saves time! What frameworks are available

- JUnit for java code
- many 'x'Unit for other programming languages (also HTTPUnit)
- closer to cocoon: XMLUnit, test documents or document fragments on equality, xpath expressions, xslt results. Need to be written in Java

What about Cocoon? Decided to build a framework for testing in cocoon, based on XSLTUnit, a stylesheet by Eric van der Vlist. Use it to transform a test-document and comparing the result to reference-output.

XSLTUnit requires one XSL extension, exsl:node-set().

Requirements

Testing framework and tests should be transparent to your application, don't interfere with your application. e.g. create a '_test' dir in each sitemap location. When you deploy to a live server, the deployment script removes each _test directory.

Little overhead code should be needed if you write a test. Just the input-file, the stylesheet, and the reference output, plus the comparison you want to make.

Setting up

Code for unit test module will be available real soon now. Mounted in the main cocoon sitemap, or via the mount table if you want to keep it out of there. A testable component must be mounted via the mount-table.xml and have a 'test' attribute. Create directory _test, and add sitemap.
=> example/_test/sitemap.xmap Create a test directory for a sitemap in your application. Create a test suite, consisting of test stylesheets, test inputs and reference outputs.

Ready to run! Nico shows the results of the XSLT Unit tests. Raw results (XML) are transformed into a nice report listing % passes and failures, and detailed description of failures.

Sample setup

```
my_component
-> sitemap.xmap
-> foo.xsl — bar.xsl
-> _test
-> sitemap.xmap
-> foo.xsl
-> 01-input.xml — 01-test.xsl
-> 02-input.xml — 02-test.xsl
-> ...
-> bar.xsl
-> ...
```

Example: resource browser

Represent a tree as nested <resource> elements. Recursive calls to a pipeline, one for each level. Pipeline uses xsl transform and cinclude for recursion. Create a test sitemap. Create a test, don't start with stylesheet, but with test input and expected output. Write a test stylesheet that does a couple of things:

- import xsltunit.xsl
- import the stylesheet you want to test
- reference output may be included in the test stylesheet.

Add another test, run, fail, refactor.

You can use parameters in your tests.

Additional features

- test output in separate document
- test pipeline fragments
- mount tests for sub-sitemaps

What's next

- automatic generation of test code
- Make a testing block?
- Tests for flowscript?
- User documentation
- Tests for framework itself

How to name the test framework

- CUTE? Cocoon Unit Testing Environment
- COURT? (missed it)
- Code will be available via the wiki (<http://wiki.apache.org/cocoon/NicoVerwer>)

Better Location?