DebuggingWithViews

Overview

What do you do when a cocoon page doesn't work? When debugging a cocoon page, you'll often want to view the results at various places in the transformation pipeline in order to identify the step that went wrong. Here's a technique that allows you to do this without having to modify your pipeline logic. This is an especially important point, as it allows you to easily apply this technique to an existing cocoon application with minimal rewriting and zero modification of pipeline logic. ILinKuo

Note: if you're debugging a form, make sure you read all the way to the end.

Add Debugging Views to Sitemap

We make use of a gem in cocoon called Views. Views allow you to label parts of a pipeline where you would like to terminate the pipeline and specify an alternate serialzation. It's easier to demonstrate than to explain.

First, in your sitemap.xmap, find the <map:views> element at add the following views for debugging:

```
<map:views>
  <!-- debugging views: add these to your views -->
  <!-- make sure you comment out debugging views before deployment -->
  <map:view from-position="first" name="first">
    <map:serialize type="xml"/>
  </map:view>
  <map:view from-position="first" name="firsthtml">
    <map:serialize type="html"/>
  </map:view>
  <map:view from-position="first" name="firsttext">
    <map:serialize type="text"/>
  </map:view>
  <map:view from-label="debug1" name="debug1">
    <map:serialize type="xml"/>
  </map:view>
  <map:view from-label="debug2" name="debug2">
    <map:serialize type="xml"/>
  </map:view>
  <map:view from-label="debug3" name="debug3">
    <map:serialize type="xml"/>
  </map:view>
  <map:view from-label="debug4" name="debug4">
    <map:serialize type="xml"/>
  </map:view>
  <map:view from-position="last" name="last">
    <map:serialize type="xml"/>
  </map:view>
  <map:view from-position="last" name="lasthtml">
    <map:serialize type="html"/>
  </map:view>
  <map:view from-position="last" name="lasttext">
    <map:serialize type="text"/>
  </map:view>
</map:views>
```

Prepare Pipeline for Debugging

Next, you'll add labels to your pipeline. Let's take the following 5-stage pipeline for example.

```
<map:match pattern="getReport.htm">
  <map:generate src="source.xml"/>
  <map:transform src="prepareSQL.xsl" type="xslt"/>
  <map:transform type="sql"/>
  <map:transform src="displayResult.xsl" type="xslt"/>
  <map:serialize type="html"/>
</map:match>
```

Add labels to the middle three stages of the pipeline as follows *(Note: labels are not needed for the first and last stages of the pipeline because we've defined the views using from-position="first" and "last", respectively)*

```
<map:match pattern="getReport.htm">
  <map:generate src="source.xml"/>
  <map:transform src="prepareSQL.xsl" type="xslt" label="debug1"/>
  <map:transform type="sql" label="debug2"/>
  <map:transform src="displayResult.xsl" type="xslt" label="debug3"/>
  <map:serialize type="html"/>
</map:match>
```

Notice that your pipeline logic has not been altered in any way. All you've done is add labels to your pipeline. **You're now ready to go!**

Using Your Debugging Views

Now if your cocoon page `www.yourSite.com\getReport.htm` gives you an error, you can just append a cocoon-view URL parameter to the page URL to view the results at each stage of the pipeline. To be more concrete, typing:

```
www.yourSite.com\getReport.htm?cocoon-view=first
www.yourSite.com\getReport.htm?cocoon-view=debug1
www.yourSite.com\getReport.htm?cocoon-view=debug2
www.yourSite.com\getReport.htm?cocoon-view=debug3
www.yourSite.com\getReport.htm?cocoon-view=last
```

allows you to view the xml at each of the five stages of the pipeline. **That's it!**

However, if you're debugging a form....

Using Cocoon-views to Debug Forms

If you're debugging the results of a form submission, you can't just add *"?cocoon-view=debug1"* to the URL, because the form submits to its action attribute. Here's what I do:

1. Browse to the submission form and save it as html on the desktop.
2. Open up the saved html and add `<input type="text" name="cocoon-view" value="debug1">` anywhere in between the `<form></form>` tags.

This adds a request-parameter named "cocoon-view" into the form which allows you to test your page without altering your cocoon application.

Comment Out Debugging Views Before Deployment

Make sure you don't enable these debugging views on your production application, as it is a **security risk**. Before deploying, you can disable all your debugging views by commenting them out in `<map:views>`.

I hope this tip helps! Feel free to modify this page.

Convenience Mozilla/IE bookmarklets

can be found Bookmarklets