

TowardsACocoonReference

Rationale

I see a big hole in the current Cocoon documentation, and I would like to work towards filling it. The issue as I see it is that, although there is a plethora of working samples and snippets, I have found no documentation describing the Cocoon system that is *comprehensive*, *current*, and *detailed*. By far the most frustrating aspect of learning Cocoon over the last few days has been that, although I have a 500m view of the components that make up a sitemap through the User Guide, and although I have detailed descriptions of certain components in various places (often as Javadocs!), there is a tremendous impedance mismatch between the two, and I must often revert to digging through code and the complex sitemaps that come with the Cocoon distribution to find what I need.

What I would really like is a *reference manual* that I as a Cocoon user can have at my fingers as I'm hacking up my website; something like a class reference that can guide me through the details of a Cocoon project quickly and efficiently once the "rubber hits the road," so to speak.

One might argue that the User's Guide be extended to fill in this gap. Yet I like the somewhat conversational and high-level presentation in the User's Guide, and I feel that would be sacrificed if the User's Guide were to become too rich in details. I feel the User's Guide serves a different audience and a different purpose, and should stay that way.

One might also argue that extending the Developer Guide would be an option. As a Cocoon user, not a developer, I disagree strongly with this. Learning and using the XML-based tools that are the bread and butter of site development is great; delving into the backend is quite another matter. Keeping the developer documentation separate is a Good Thing for me.

Proposed Outline

As a newcomer to Cocoon, I can't pretend to have a complete idea of what a reference manual for Cocoon would look like. Here's a starter culture:

- Running Cocoon
 - As a standalone servlet (using Jetty)
 - As a Tomcat servlet
 - As a command line application
- Configuration Reference
- Sitemap Reference
 - Components
 - Resources
 - Pipelines
 - etc.
- Component Reference
 - Readers
 - Generators
 - Transformers
 - Serializers

I expect the components would make the bulk of the documentation, so should be broken out into their own section. Of course, the "standard" components would be the targets for the reference, much as they are in the User's Guide.

— [OwenSmith](#) 24-Jun-2004