Overview of Apache Cocoon (2.1 legacy document)

Table of contents

Table of contents

Warning:
This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. What is Apache Cocoon

Cocoon is an XML publishing framework. It allows you to define XML documents and transformations to be applied on it, to eventually generate a presentation format of your choice (HTML, PDF, SVG, ...).

Cocoon also gives you the possibility to apply logic to your XML files (so that the XML pipeline can be dynamic).

The User documentation and especially Concepts will help to understand Cocoon.

2. Examples and demonstration applications

There are a whole suite of sample applications to demonstrate the power of Cocoon. These samples are available from the "welcome" page after you have downloaded, built, and installed the distribution. Each example portrays a different aspect of the vast capabilities of Cocoon ... http://localhost:8080/cocoon/

With the 2.1 version, http://localhost:8080/cocoon/ goes directly to the documentation, while http://localhost:8080/cocoon/samples/ is the Samples.

It will greatly assist your understanding of Cocoon to investigate behind-the-scenes, to find out how each sample is processed. Do this by looking at the actual XML documents provided in the distribution at src/webapp/samples/ and by consulting each sitemap to see the processing steps that are defined.

3. Overview of XML document processing

This section gives a general overview of how an XML document is handled by Cocoon. See also the document Understanding Cocoon for explanation of the separation of content, style, logic and management functions.

3.1. Pipeline

Cocoon relies on the pipeline model: an XML document is pushed through a pipeline, that exists in several transformation steps of your document. Every pipeline begins with a generator, continues with zero or more transformers, and ends with a serializer. This can be compared to the "servlet-chaining" concept of a servlet engine. We'll explain the components of the pipeline now in more detail.

3.1.1. Generator

The Generator is the starting point for the pipeline. It is responsible for delivering SAX events down the pipeline.

The simplest Generator is the FileGenerator: it takes a local XML document, parses it, and sends the SAX events down the pipeline.

The Generator is constructed to be independent of the concept "file". If you are able to generate SAX events from another source, you can use that without having to go via a temporary file.

3.1.2. Transformer

A Transformer can be compared to an XSL: it gets an XML document (or SAX events), and generates another XML document (or SAX events).

The simplest Transformer is the XalanTransformer: it applies an XSL to the SAX events it receives.

3.1.3. Serializer

A Serializer is responsible for transforming SAX events to a presentation format. For actors looking at the back of the pipeline, it looks like a static file is delivered. So a browser can receive HTML, and will not be able to tell the difference with a static file on the filesystem of the server.

We have Serializers for generating HTML, XML, PDF, VRML, WAP, and of course you can create your own.

The simplest Serializer is the XMLSerializer: it receives the SAX events from up the pipeline, and returns a "human-readable" XML file.

1. Comments

add your comments