BricksCms

This page is meant to give up-to-date information about the bricks-cms example application.

The application has been presented at the GetTogether 2005, see http://www.cocoongt.org/speakers/bertrand.html and the presentation text, drawings and audio recording at http://www.cocoongt.org/Slides-and-recordings.html

If you like drawings, the "what you need to know about Cocoon" presentation at the GT 2004 also includes a diagram of a typical Cocoon application, it is available under "Material from events" at http://cocoon.apache.org/mirror.cgi, under gt2004/presentations/gt2004-bertrand-...

What is bricks-cms?

Bricks-cms is NOT a real CMS, it's only an example application meant to demonstrate how to work with database and java components in Cocoon 2.1.x, and how to implement CRUD applications. And even if you're creating pure publishing applications (XSLT, sitemaps but little or no Java code), Bricks shows a simple ant-based build system which lets you cleanly separate your code from Cocoon's.

It was called "cms" because a content management system is a good use case for Cocoon.

But it's no competition for the "real" Cocoon-based CMSes out there - if you're looking for a CMS there are several systems based on Cocoon, but this one is only really a toy.

Overview

I found this project to be very useful for illustrating Cocoon, as it is mostly stripped down to the basics. The distro has a lot of stuff in it which can be distracting at times.

Under src, we have 4 directories: build-targets, cocoon, java, misc. build-targets has some Ant targets which are included by the main build.xml. misc contains some important files for the auxiliary projects that bricks relies on (derby, hivemind, ojb). Let's look at the other 2 dirs.

Java

All the Java code is organized under org.apache.coocon.apps.bricks.cms. The packages are:

- componentmanager: Component initialization & management; will be called by servlet container when context is created/destroyed
- data: CMS application data (documents)
- objectstore: Interface between OJB & objects under "data"; simple create & search operations
- ojb: JDBC bridge; database connection & transaction management
- util: Exception classes, etc.
- wikiparser: Interface & implementation of Wiki parser (actually uses the textile4j parser)

note: I don't think I explained objectstore & ojb very well here.

Cocoon

We have 3 dirs here: config, webapp, xconf. config holds the properties used to build Cocoon from the distribution; it would be similar to local.build.properties under Cocoon. xconf holds patches against the default Cocoon files that would be found under WEB-INF (see the readme for how to use this). Let's look at the webapp:

Webapp

(to be continued)

News

- October 7th, 2005: code snapshot uploaded, will be available on our download mirrors in the next few hours.

Current status

A snapshot of the current code is available at http://cvs.apache.org/dist/cocoon/bricks-cms/ .

The latest code is available from our Subversion repository at http://svn.apache.org/repos/asf/cocoon/whiteboard/example-apps/bricks-cms/

The version number is shown in the title of the admin interface homepage at http://localhost:8888/bricks-cms/admin/homepage .

See the README.TXT and STATUS.TXT files there for details.

Testing

Until now the app has only been tested in the following enviromnents:

- Cocoon 2.1.7, macosx 10.3.8, JDK 1.4.2, ant 1.6.5
- Cocoon 2.1.7, Linux 2.6.10, JDK 1.5, tomcat 5.5 (thanks to Aurélien Dehay!)
- Cocoon 2.1.7, Windows XP, JDK 1.4.2, ant 1.6.2
- Cocoon 2.1.8, Windows XP, JDK 1.5.0, ant 1.6.5
- Cocoon 2.1.8, Linux 2.6.15, JDK 1.4.2_08, Ant 1.7alpha
- Cocoon 2.1.9, Linux 2.6.11, JDK 1.5.0_04, tomcat 5.5
- Cocoon 2.1.9, Windows XP, JDK 1.5.0_08, Jetty, Eclipse 3.2
- (add your environment here...)

Testing information is found in the README.TXT file.

If someone tests it on other platforms, please report your findings here (or send patches, or commit your corrections)!

What's next?

The app will be completed before the GT next week, but I'm not planning on adding much code, it has to stay minimal to make it easy to explore the source code.