

Cocoon Source Resolving (2.1 legacy document)

Table of contents

1 Comments.....3

Table of contents

1 Differences between Cocoon and Avalon Excalibur Source Resolving.....	3
1.1 The Interfaces.....	3
1.2 Caching.....	3
1.3 XMLizable.....	3

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. Differences between Cocoon and Avalon Excalibur Source Resolving

This story tries to list the changes between the two source resolving approaches. Originally, the source resolving was developed in the Cocoon community. As the community recognized that it had a more common value, the source resolving was donated to the Avalon Excalibur project and there it was refined and redesigned. Now the way back takes place by reintegrating the Avalon Excalibur Source Resolving into Cocoon and deprecating the now obsolete Cocoon Source Resolving.

1.1. The Interfaces

Besides the package names the main difference is that the Cocoon Source object is XMLizable, that means it can directly send SAX events. The AE Source object is not XMLizable, which is more correct as a protocol only describes the way of transport but not the format of the data which is transferred. Therefore it provides a getMimeType() method.

1.2. Caching

The Cocoon Source object returns a last modified time stamp which is very limiting. For example a source could have an expires date or something like that. The AE Source returns a validity object which is capable of all these validity checkings. So the information used for validity checking is provided by the source object and not maintained by the client of the source as it is currently in Cocoon.

1.3. XMLizable

As we in Cocoonland are dealing a lot with XML, a Source implementation can still implement the XMLizable Interface (from AE and not from Cocoon!) to directly provide SAX events. The Cocoon SourceUtil class has a method toSAX() which gets a Source object and tests if it implements the XMLizable interface. If not the new AE XMLizer is used which is a component that can convert an input stream with a given mime-type to SAX events. This component is configurable and extensible.

1. Comments

add your comments