

JcrSource

The Cocoon JCR (JSR-170) Source

Cocoon comes with a JCRSource in 2.1.8-dev. You can find it in the JCR block ([2.1.x source code](#), [2.2 source code](#))

This page is meant as an overview to get started with an integration of Cocoon and a JCR repository.

Another JCR Source that maps XML to JCR and allows for full XPath (only with Jackrabbit) was developed for Mindquarry: [mindquarry-jcr svn](#). A bit of documentation can be found in the [Mindquarry architecture guide](#).

What is the JCRSource?

The JCRSource is a writeable, traversable Cocoon source. It is usually bound to the jcr: protocol. In other words: You can use src="jcr:..." anywhere in Cocoon where you would be able to use other protocols such as src="cocoon:..." or src="context:...".

This provides an easy way for Cocoon to access content that resides in a Java Content Repository (JCR).

What kind of Content can be fetched from a JCR?

A JCR is not limited in terms of the type of data it can hold. You can store XML, but also pictures, plain ASCII text or video files in a JCR repository. Therefore you must keep in mind to combine the JCRSource with an appropriate generator. If you are accessing plain text for example, you need to use the [TextGenerator](#) in your sitemap. The JCR source will not convert plain text in the repository into XML for you.

There is also no mime type magic implemented, though the JCR repository can contain mime type information. Theretically it would be possible to lookup the mime type of a node in the repository and choose a generator based upon that, but this is just not implemented yet. (And it could be debated how useful this would be or not.)

How does the content make it into the JCR in the first place?

The JCR source is writeable, but unless you use a Cocoon based CMS such as for example Lenya, you will probably not use the JCR source to write to the repository. JCR is a young technology indeed and we will most likely see very nice tools to import content into a JCR repository very soon. Cocoon will unfold its strengths when it comes to rendering the content into different channels.

Configuring the JCRSource

With the current implementation, one single instance of Cocoon is limited to connecting to a single repository. In order to configure the JCRSource, you need to edit two different sections in cocoon.xconf.

The javax.jcr.Repository role

This is a component configuration (note: not a component-instance!) which supplies the following parameters:

- credentials: The credentials needed to log into the repository
- jaas: The JAAS configuration file
- home: The file system directory where the repository resides (note: you can use protocols such as context: here)
- configuration: The path of the repository.xml file. Refer to the Jackrabbit documentation for the syntax of that file.

Here is an example of the javax.jcr.Repository role configuration:

```
<component class="org.apache.cocoon.jcr.JackrabbitRepository" logger="jcr" role="javax.jcr.Repository">
  <credentials login="anonymous" password="" />

  <jaas src="context://samples/blocks/jcr/jaas.config"/>

  <home src="file:/usr/local/src/jackrabbit/contrib/examples/target/repo"/>
  <configuration src="file:/usr/local/src/jackrabbit/contrib/examples/target/repo/repository.xml"/>

</component>
```

The jcr JCRSource component instance

This sections contains configuration information that the JCR source needs in order to map the file system (path name) metaphor of the jcr: protocol to the repository.

```
<component-instance class="org.apache.cocoon.jcr.source.JCRSourceFactory" name="jcr">
  <folder-node new-file="nt:file" new-folder="nt:folder" type="rep:root"/>
  <folder-node new-file="nt:file" new-folder="nt:unstructured" type="nt:unstructured"/>

  <file-node content-path="jcr:content" content-type="nt:resource" type="nt:file"/>
  <file-node content-ref="jcr:content" type="nt:linkedFile"/>

  <content-node type="nt:resource"
    content-prop="jcr:data"
    mimetype-prop="jcr:mimeType"
    lastmodified-prop="jcr:lastModified"
    validity-prop="jcr:lastModified"/>

  <content-node type="nt:unstructured"
    content-prop="jcr:xmlcharacters"
    mimetype-prop="jcr:mimeType"
    lastmodified-prop="jcr:lastModified"
    validity-prop="jcr:lastModified"/>

</component-instance>
```

Some insights to this can be found here:

<http://www.mail-archive.com/users%40cocoon.apache.org/msg28307.html>

JCR links

- The Apache Jackrabbit Project (<http://incubator.apache.org/jackrabbit>)
- Karma Repositorybrowser (<https://karma.dev.java.net/source/browse/karma/repositorybrowser/>, read the Jackrabbit Mailing List archives for details on state and capabilities)