OverviewSitemapEntryPoints

Sitemap Entry Points in the Default Publication

Each Lenya publication is in theroy free to define its URI space by setting up an individual sitemap.xmap in its lenya/pubs/<your-pub> directory. But in practice, many publications are based on the *Default Publication* that comes with Lenya.

Even if you plan an entirely different structure for your publication it might still be worth taking the time to understand the typical mechanism in place inside a Lenya publication. If you want to make use of the Lenya framework at all to provide features such as the workflow (separation of authoring and live area) or the site navigation framework introduced by Lenya to your site editors, you will need to employ similar (if not the same) mechanisms.

What is a sitemap entry point?

The term *pipeline* which is used a lot in Cocoon suggests a construct that is quite linear, such as a gas or oil piepline made of tubes. It has one end to feed stuff into that will exit the pipleline on the other end. But the Cocoon pipeline that is implemented in Lenya is a bit more as in Nintendo's Super Mario video game. It has lots of junctions and sub-pipelines.

If your are setting up simple publications you will probably never have to touch the structure of sitemaps and sub-sitemaps at all, but you will get away with the changes to the *Default Publication* explained in the ApplicationLookandFeelHowto. But if you dig deeper and start implementing your own doctypes or if you are interested in introducing advanced concepts into your publication such as wiki notation or local edit you will need to understand the pipelines in sub-sitemaps available to you and how to call them in the right place.

The entry points are usually called from somewhere else in the any level of the sitemap using the cocoon:// subprotocol. Usually you cannot call them directly from your browser, which would make sense for debugging purposes, but tests and the need for certain variables in a higher level (parent) request context renders this impossible unfortunately.

publication-sitemap: lenya-document-*/*/*/**.xml

Purpose: Render the XML file to XHTML

- **Parameter {1}**: {rendertype}
- **Parameter {2}**: {area} (authoring | live)
- **Parameter {3}**: {doctype}
- **Parameter {4}**:

The requested document specified by the document path in {4} might be arbitrary XML. It does not necessarily have to be XHTML at all, especially if you are using your own doctypes. The `lenya-document`-pipeline is responsible for rendering the XML document into XHTML.

This is achieved by calling into the doctype.xmap-pipeline, which in the *Default Publication* uses the Cocoon FileGenerator to input the file into the pipeline and apply the document type specific XSLT stylesheet which needs to follow the _doctype_2page.xsl naming convention.

publication-sitemap: lenyabody-*/*/*/*/**

Purpose: Aggregate site navigation (breadcrumbs, tabs and menus) and page content

- **Parameter {1}**: {rendertype} ( )
- **Parameter {2}**: {publication-id}
- **Parameter {3}**: {area} (authoring | live)
- **Parameter {4}**: {doctype}
- **Parameter {5}**: {url}

Relies on: `lenya-document-*/*/*/**.xml`

See also

- AddingCustomDocType