Link Management

Table of contents

1 Introduction....2
2 Syntax for Internal Links....2
3 Specifying the Resource Type Format....2
4 Converting UUID-based URLs to Web Application URLs....3
4.1 Specifying an Extension....3
5 Converting Web Application Links to Servlet Container or Proxy Links....3
6 URLs in CSS Files....4

1. Introduction

Link Managements deals with internal links, i.e. documents that refer to other documents within the same publication. In Lenya 1.2, links are rewritten each time the URL of the target document changes (see section Link Management (../../../docs/1_2_x/components/link-management/link-management.html) in the Lenya 1.2 documentation).

In Lenya 2.0, documents are identified and referenced using UUIDs. The UUID of a document never changes, that's why the internal links don't have to be rewritten when a document is moved in the site structure.

2. Syntax for Internal Links

Internal links have the following syntax:

```
lenya-document:{uuid}[,lang={…}][,pub={…}][,area={…}][,rev={…}][{queryString}]
```

- The `{uuid}` value is usually present, but optional. This enables you to create links to "the russian translation of this page" (`<a href="lenya-document:,lang=ru">...`). The language selector module makes use of this feature. Other conceivable uses are links to the current document in another area or revision.
- The parameters `lang`, `pub`, `area`, and `rev` are optional. If omitted, the corresponding values of the current context and the latest revision are used.
- The optional query string is not evaluated when resolving the link target. It can be used to provide information for subsequent processing steps (e.g., the `UuidToUrlTransformer`) and to add request parameters to the resulting HTTP link URL. The query string is separated from the link URL using a question mark.

Some link URI examples:

- `lenya-document:031b21e0-d898-11db-b5f3-e5afac0e35a0` - document in same language
- `lenya-document:031b21e0-d898-11db-b5f3-e5afac0e35a0,lang=de` - document in different language
- `lenya-document:031b21e0-d898-11db-b5f3-e5afac0e35a0,area=trash` - document in different area
- `lenya-document:031b21e0-d898-11db-b5f3-e5afac0e35a0?uuid2url.extension=rss` - HTML link should use a particular extension
- `lenya-document:031b21e0-d898-11db-b5f3-e5afac0e35a0?lenya.usecase=bxe.edit` - HTML link with request parameter

3. Specifying the Resource Type Format

Sometimes you want to specify the format of the link target. This can be done using the `format` parameter. A typical use case is the inclusion via CInclude.

Imagine you're converting a collection of documents to a list of teasers. The collection source might look like this:

```
<col:collection>
  <col:document uuid="031…"/>
  <col:document uuid="a5r…"/>
  …
</col:collection>
```

You could use the following XSLT template to generate the CInclude statements for the teaser list:

```
<xsl:template match="col:document">
  <ci:include src="lenya-document:{@uuid}?format=teaser"/>
</xsl:template>
```

When the resulting XML is processed by

```
<map:transform type="cinclude"/>
```

the teaser version of each document will be included.

4. Converting UUID-based URLs to Web Application URLs

The `UuidToUrlTransformer` is responsible for translating `lenya-document:` links into the corresponding web application links. The context path or proxy settings are not considered yet, this is the responsibility of the ProxyTransformer (see below). So don't forget to apply the ProxyTransformer after the UuidToUrlTransformer.

If your source document contains a link like this:

```
<a href="lenya-document:031…">News</a>
```

and your pipeline contains a call to the UuidToUrlTransformer:

```
<map:transform type="uuid2url"/>
```

the resulting HTML link will look like this:

```
<a href="/default/authoring/news.html">News</a>
```

4.1. Specifying an Extension

The UuidToUrlTransformer recognizes the query string parameter `uuid2url.extension`. You can use it to force a specific extension. For instance

```
<a type="application/rss+xhtml"
href="lenya-document:031…?uuid2url.extension=rss">RSS Feed</a>
```

will result in

```
<a type="application/rss+xhtml" href="/default/authoring/news.rss">RSS Feed</a>
```

5. Converting Web Application Links to Servlet Container or Proxy Links

The `ProxyTransformer` converts all web application links to final links by adding the servlet context path or the proxy URL. For instance, if you run your Lenya servlet in Tomcat under the context path *lenya14*, transforming

```
<a href="/default/authoring/news.html">News</a>
```

with the ProxyTransformer

```
<map:transform type="proxy"/>
```

will result in

```
<a href="/lenya14/default/authoring/news.html">News</a>
```

If you have declared a proxy for the authoring area, the resulting link might look like this:

```
<a href="http://cms.mysite.com/news.html">News</a>
```

If you have configured the proxy transformer to use relative URLs, the link will be converted to something like this:

```
<a href="../../news.html">News</a>
```

Typically, the ProxyTransformer is applied right after the UuidToUrlTransformer. For more information on the ProxyTransformer, consult the API documentation.

The `ProxyModule` provides the same functionality in sitemaps. It takes a URL as parameter and rewrites it in the same way as the ProxyTransformer:

```
<map:parameter name="url" value="{proxy:/default/authoring/news.html}"/>
```

6. URLs in CSS Files

URLs in CSS files are converted to valid links automatically. In your CSS, you can just write

```
div.news { background: url('/mypub/live/images/news.png') left top no-repeat }
```

and, according to your proxy settings or context path, this will end up as one of these:

```
div.news { background: url('/lenya14/mypub/live/images/news.png') left top no-repeat
}
div.news { background: url('http://mysite.com/images/news.png') left top no-repeat }
div.news { background: url('../images/news.png') left top no-repeat }
```

This operation cannot be performed by our standard XSLT transformer, because a CSS file is not well-formed XML. The Chaperon (http://chaperon.sourceforge.net) transformer is used instead.