

Meta Data

Table of contents

| | |
|-----------------------------------|---|
| 1 Introduction..... | 2 |
| 2 Example..... | 2 |
| 3 Implementation..... | 2 |
| 3.1 Create meta data..... | 2 |
| 3.2 Display/modify meta data..... | 3 |

1. Introduction

In 1.4.x meta data is kept apart from the xml document but in the same directory (index_{lang}.xml.meta). There are different types of meta data.

1. The default is based on the [Dublin Core](http://dublincore.org) (<http://dublincore.org>) standard (like in 1.2.x).
2. Besides we introduced internal (used by lenya internally and not exposed to the user)
3. and custom metadata (added from the developer to enable user to maintain their extra informations about the document)

There are basically two use cases as example:

1. Create meta data
2. Display/modify meta data

2. Example

A typical sample for meta data may be the following:

```
<lenya:document xmlns:lenya="http://apache.org/cocoon/lenya/page-envelope/1.0"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <lenya:meta>
    <lenya:custom>
      <lenya:myCustom>My custom metadata</lenya:myCustom>
    </lenya:custom>
    <lenya:internal>
      <lenya:resourceType>xhtml</lenya:resourceType>
      <lenya:contentType>xml</lenya:contentType>
    </lenya:internal>
    <lenya:dc>
      <dc:title>Some Lenya Features quickly explained</dc:title>
      <dc:creator>Lenya Development Team</dc:creator>
      <dc:subject>Lenya Features</dc:subject>
      <dc:description>Explains some Lenya features</dc:description>
      <dc:publisher>Apache Software Foundation</dc:publisher>
      <dc:contributor/>
      <dc:date>2005-01-31</dc:date>
      <dc:type/>
      <dc:format/>
      <dc:identifier/>
      <dc:source/>
      <dc:language>en</dc:language>
      <dc:relation/>
      <dc:coverage/>
      <dc:rights>All rights reserved</dc:rights>
    </lenya:dc>
  </lenya:meta>
</lenya:document>
```

3. Implementation

Like nearly all new modules/functionality the meta data usecases are following the new **fallback** concept. Meaning you are using the core contracts as long you are **not** overriding them with your own implementation. To override a core implementation you just need to place your custom implementation to the right path in you pub and lenya will try to pick it up from there.

3.1. Create meta data

Upon creation of a document a set of sample meta data is presented in the creation form. This values are partially filled in by the user (subject, description, etc.) and partly by the system (creator, creation data). This is done with the `site.create` usecase (lenya.usecase=site.create).

We activated that all custom meta data will be picked up from the form and saved to the meta data doc. To tell lenya that you

want as well create a set of custom meta data, you need to modify your implementation of the `create.jx` form.

Custom implementation of `create.jx`

An **example** of an implementation can be found in `{ $default-pub } / lenya / usecases / site / create . jx`. Just change it and see what comes out. BTW if you need it in your custom pub just mind the path. ;-)

3.2. Display/modify meta data

The display of meta data is handled by the usecase `tab.meta`. We activated that all custom meta data are passed to the form. However you will need to add the fields you want to edit in your own implementation of the `meta.jx`.

Custom implementation of `meta.jx`

An **example** of an implementation can be found in `{ $default-pub } / lenya / usecases / tab / meta . jx`. If you need it in your custom pub just mind the path.