Mulitilingual Document handling

Table of contents

1. Introduction

Mulitilingual Document handling deals with documents of different languages. This affects virtually every part of Lenya, namely many aspects of work flow such as creation, publication, deactivation, deletion. It also affects access control, meta data, etc.

The multilingual capabilities are not a mandatory feature of Lenya. You can easily create monolingual publication by simply replacing a few properties in `publication.xconf`.

The multilingual capabilities are deeply ingrained in the core of Lenya and as such a conceptual overview has to delve into many areas of Lenya. The main parts of the multilingual capabilities are as follows:

Site tree
The site tree (../publication/siteTree.html) contains information about documents. It has been expanded to also contain information about different language versions of the same document. Different language versions of the same document share the same `document-id`.

Default language
A publication has a default language. This is the language in which the document is created. However it is not necessarily the case that there is always a document version in the default language as the user is allowed to remove any language version even the one for the default language.
If you specify an URL without a language suffix you are redirected to the default language document.

publication.xconf
This is where the available languages and the default language are specified.

page-envelope
The page envelope (../publication/pageenvelopemodule.html) exports document specific information such as document-id via an input module. Naturally language specific information for the current document is also available via the page envelope.

Redirection
If you request a document `foo.html` in a multilingual publication you will internally be redirected to `foo_en.html` if "en" is the default language.

Special cases
Some special care has to be taken when removing language versions. The current implementation does not allow the user to remove all language versions of a document. There always has to be at least one language version.

2. Implementation

2.1. Site tree

The site tree contains nodes for each document. Each node can contain multiple `label` nodes for each language version of the document.

```
<node id="doctypes">
        <label xml:lang="de">Dokumenttypen</label>
        <label xml:lang="en">Document types</label>
        <node id="1column">
                <label xml:lang="de">1 Spalte</label>
        </node>
</node>
```

The classe `SiteTree` and `SiteTreeNode` provide interfaces to the nodes and the corresponding label nodes.

2.2. publication.xconf

The `languages` node in `publication.xconf` defines the languages that are available for this publication and also defines the default language.

```
<languages>
  <language default="true">de</language>
  <language>en</language>
</languages>
```

This information is available via the `Publication` class (`getDefaultLanguage()` and `getLanguages()` methods).

2.3. page-envelope

The page-envelope input module (../publication/pageenvelopemodule.html) exports language related information, namely `default-language`, `document-label`, `document-language`, `document-languages` and `document-languages-csv`.

2.4. Redirection

The `DefaultDocumentBuilder` which implements the `DocumentBuilder` interface, takes care of redirecting language agnostic URLs (such as `foo.html`) to language aware links (`foo_en.html`), i.e. language agnostic links are redirected to the language aware link with the default language.

2.5. Addition and removal

More language versions are added with the "Create new language version" screen. Existing language versions can be removed with the "Remove Language" screen. These screens are generated by `create-language.xsp`, `create-language.xsl`, `removelabel.xsp` and `removelabel.xsl`.

The creation of a new language version is done via the `create-language` target which uses the `InsertLabelTask` ant task to insert a label in the site tree.

The removal of a language version is done via the `remove-language` target, which uses the `RemoveLabelTask` ant task (which removes the label from the site tree).

3. Involved classes, XSPs and XSLTs

The following classes, XSPs and XSLTs are involved in multilingual document handling:

org.apache.lenya.cms.publication.SiteTree and org.apache.lenya.cms.publication.SiteTreeNode
These interfaces provide acces to the label nodes.
org.apache.lenya.cms.publication.Publication
Provides access to the default language and to all available languages.
org.apache.lenya.cms.publication.PageEnvelope
Exports language related information such as `default-language`, `document-language`, etc.
org.apache.lenya.cms.publication.DefaultDocumentBuilder
Builds a document with the given language. The default implementation also takes care of redirecting `foo.html` to `foo_en.html`.
$publication-id/lenya/content/authoring/create-language.xsp, $publication-id/lenya/xslt/authoring/create-language.xsl
Generate the "Create new language version" screen.
src/webapp/lenya/content/authoring/removelabel.xsp, src/webapp/lenya/xslt/authoring/removelabel.xsl
Generate the "Remove Language" screen.
org.apache.lenya.cms.ant.InsertLabelTask
Ant task to add a label to the site tree.

org.apache.lenya.cms.ant.RemoveLabelTask
Ant task to remove a label from the site tree.
$publication-id/config/tasks/targets.xml
Defines the `create-language` and the `remove-language` target which handle the addition and removal of language versions of a document.