The sitetree Module

Table of contents

1 Introduction

The `sitetree` module provides a tree-based site manager.

2 Generating Navigation Elements

One purpose of the sitetree module is to simplify and standardize the generation of navigation widgets like menubars, tabs, and breadcrumb paths. It allows to

- use pre-defined XHTML navigation components and present them with a custom CSS,
- override these components in a publication whereby the logic can be reused, and
- create and integrate new publication-specific navigation components using the same scheme.

3 Architecture

The navigation element mechanism consists of the following components:

`modules/sitetree/sitemap.xmap`
The sitemap where the navigation components are generated.
`modules/sitetree/xslt/navigation/*.xsl`
The default navigation components that are shipped with the sitetree module.
`{your-publication}/lenya/modules/sitetree/xslt/navigation/*.xsl`
Custom navigation components of the publication.

4 Loading a Navigation Widget

You can load a navigation widget from the following URI:

```
cocoon://modules/sitetree/{widget}/{pub-id}/{area}/{default-lang}/{lang}{path}.xml
```

The URI steps are:

`<widget>`
The type of navigation widget to load (tabs, menu, etc.). This will be the name of the XSLT which generates the widget.
`<pub-id>`
The publicaton ID of the current documen.
`<area>`
The area of the current document.
`<default-lang>`
The default language of the publication. We pass this value instead of determining it in the module sitemap so that the generated widget doesn't contain any dynamics which are not coded in the URL and therefore can be cached in the publication.
`<lang>`
The language of the document.
`<path>`
The path of the current document in the site structure, starting with a slash.

Here's an example navigation widget URL:

```
cocoon://modules/sitetree/tabs/mypub/authoring/en/de/foo/bar.xml
```

5 Aggregating the Navigation Widgets

In the publication sitemap (`{pub-id}/sitemap.xmap`) the navigation widgets are aggregated. Here you decide which components you want to use:

```
<!-- navigation/{1:widget}/{2:pub-id}/{3:area}/{4:def-lang}/{5:lang}/{6:path} -->
<map:match pattern="navigation-element/*/*/*/*/*/**">
  <map:generate src="cocoon://modules/sitetree/{1}/{2}/{3}/{4}/{5}/{6}.xml"/>
  <map:serialize type="xml"/>
</map:match>

<!-- /lenyabody-{1:rendertype}/{2:pub-id}/{3:area}/{4:doctype}/{5:def-lang}/{6:lang}/{7:path} -->
<map:match pattern="lenyabody-raw-*/*/*/*/*/*/**">
  <map:aggregate element="cmsbody">
    <map:part src="cocoon:/navigation-element/breadcrumb/{2}/{3}/{5}/{6}/{7}"/>
    <map:part src="cocoon:/navigation-element/tabs/{2}/{3}/{5}/{6}/{7}"/>
    <map:part src="cocoon:/navigation-element/menu/{2}/{3}/{5}/{6}/{7}"/>
    <map:part src="cocoon:/navigation-element/search/{2}/{3}/{5}/{6}/{7}"/>
    ...
    <map:part src="{resource-type:format-xhtml}?rendertype={1}"/>
  </map:aggregate>
  <map:serialize type="xml"/>
</map:match>
```

6 Developing Navigation Widgets

The following contracts define the development of navigation components:

- A navigation widget is generated by an XSLT stylesheet which is located at
  modules/sitetree/xslt/navigation/{widget}.xsl
  for default components and
  **lenya/pubs/<publication-id>/lenya/modules/sitetree/xslt/
  navigation/{widget}.xsl**
  for publication-specific components, including the ones that override the default components.
- The default widgets produce an XHTML fragment with the top level element <div
  class="{widget}"/>.