

Publication Configuration

Table of contents

1 The publication.xml file..... 2

1 The publication.xml file

The main configuration file of a publication is `$PUB_HOME/config/publication.xml`. The namespace of this XML file is `http://apache.org/cocoon/lenya/publication/1.1`.

```
<publication xmlns="http://apache.org/cocoon/lenya/publication/1.1" >
```

The first section contains some meta data about the publication which is displayed on overview pages etc.

```
<name>Default Publication</name>
<description>This publication is a best practice, getting started publication.</description>
<!-- the @...@ macros are evaluated by ant at build time -->
<version>@lenya.version@</version>
<lenya-version>@lenya.version@</lenya-version>
<lenya-revision>@lenya.revision@</lenya-revision>
<cocoon-version>2.1.10-dev</cocoon-version>
```

In the languages section you define which content languages should be supported. Exactly one language has to be identified as the default language.

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

The following section declares service implementations which are used by the publication.

- The `template-instantiator` element references an `Instantiator` declaration in `cocoon.xconf`.
- The `path-mapper` element declares the `DocumentIdToPathMapper` implementation which is used by this publication.
- The `document-builder` element references a `DocumentBuilder` declaration in `cocoon.xconf`.
- The `site-manager` element references a `SiteManager` declaration in `cocoon.xconf`.

```
<template-instantiator name="default"/>
<path-mapper>org.apache.lenya.cms.publication.DefaultDocumentIdToPathMapper</path-mapper>
<document-builder name="default"/>
<site-manager name="tree"/>
```

The `resource-types` section declares all resource types which are used by this publication and assigns them workflow schemas. When the "Create" usecase is invoked without a parameter denoting the resource type, you can present a list of all these resource types to choose from (this is the default behaviour).

```
<resource-types>
  <resource-type name="xhtml" workflow="fallback://config/workflow/workflow.xml"/>
  <resource-type name="homepage" workflow="fallback://config/workflow/workflow.xml"/>
  <resource-type name="links" workflow="fallback://config/workflow/workflow.xml"/>
  <resource-type name="cforms" workflow="fallback://config/workflow/workflow.xml"/>
  <resource-type name="opendocument" workflow="fallback://config/workflow/workflow.xml"/>
  <resource-type name="resource" workflow="fallback://config/workflow/workflow.xml"/>
  <resource-type name="usecase" workflow="fallback://config/workflow/workflow.xml"/>
```

```
</resource-types>
```

The `modules` section declares all modules which are used by this publication. This is necessary to generate the menus accordingly and to access the `il8n` messages from these modules.

```
<modules>
  <module name="editors"/>
  <module name="xhtml"/>
  <module name="links"/>
  <module name="opendocument"/>
  <module name="cforms"/>
  <module name="homepage"/>
  <module name="resource"/>
  <module name="sitemanagement"/>
  <module name="sitetree"/>
  <module name="export"/>
  <module name="workflow"/>
  <module name="notification"/>
  <module name="svg"/>
  <module name="lucene"/>
  <module name="development"/>
  <module name="languageselector"/>
  <module name="administration"/>
  <module name="workflow"/>
</modules>
```

The optional `content-dir` element points to the parent directory of the publications `content` directory. You can use either an absolute file system path, or a path relative to the web application context (e.g. `lenya/pubs/default`).

```
<content-dir src="/home/USERNAME/data/default"/>
```

The optional `proxies` section contains the proxy configuration of this publication. For more information see [Proxying](#) (`../../docs/2_0_x/tutorials/proxy/proxying.html`) and the [tutorial](#) (`../../docs/1_2_x/tutorial/mod_proxy_and_lenya.html`).

In short the idea of the following proxy snippet in the configuration is a rewrite of links. Using proxies will prefix all absolute links (starting with `"/`) to absolute urls (the proxy host).

```
<proxies>
  <proxy area="live" ssl="true" url="https://www.host.com/ssl/default"/>
  <proxy area="live" ssl="false" url="http://www.host.com/default"/>
  <proxy area="authoring" ssl="true" url="https://www.host.com/lenya/default/authoring"/>
  <proxy area="authoring" ssl="false" url="http://www.host.com/lenya/default/authoring"/>
</proxies>
```

Since we have the concept of areas we define the proxy per area and for the different types of pages (ssl protected or not). For example if a link points to a document in the live area and this document is ssl protected then it will be rewritten to `https://www.host.com/ssl/default{$link}`. Where `{link}` is the linked document.

For all links that are not pointing to a document the global proxy rule in `cocoon.xconf/cocoon/component[@role = 'org.apache.lenya.cms.linking.GlobalProxies']` is applied (see `$LENYA_HOME/src/modules-core/linking/config/cocoon-xconf/globalproxies.xconf`):

```
<xconf xpath="/cocoon" unless="/cocoon/component[@role =
  'org.apache.lenya.cms.linking.GlobalProxies']">
  <component logger="lenya.proxy">
```

```
        role="org.apache.lenya.cms.linking.GlobalProxies"  
        class="org.apache.lenya.cms.linking.impl.GlobalProxiesImpl">  
    <!--  
      <proxy ssl="false" url="http://cms.host.com/foo"/>  
      <proxy ssl="true" url="https://cms.host.com/foo"/>  
    -->  
  </component>  
</xconf>
```

For example a link such as `/modules/myModule/somePath/some.thing` will be rewritten to `http://cms.host.com/foo/modules/myModule/somePath/some.thing`. In short `"/` will be replaced with `{proxies/@root}`.

```
</publication>
```