HowToXHTMLTemplating

How to setup XHTML Templating in a Lenya Publication (default pub)

Take a look at http://lenya.apache.org/1_2_x/components/layout/xhtml-templating.html to get info on what XHTML Templating is and reasons for using it.

Create your template

Usually your template can just be a sample of your page. When importing an existing site from dreamweaver, I usually just use one of the pages in my current site as a template for my Lenya pub. Your template needs to have container elements for the various areas of the page that need to contain page specific content.

Your template could look something like this (taken from the default pub)

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
      <link rel="stylesheet" href="/css/page.css" type="text/css"/>
      <meta content="Apache Lenya" name="generator"/>
      <title></title>
    </head>
    <body>
      <div id="page">
      <table width="100%" cellpadding="0" cellspacing="0" border="0">
        <tr>
          <td id="publication-title">
          </td>
          <td id="project-logo"><img src="/images/project-logo.png" alt="project logo"/></td>
        </tr>
      </table>
      <div id="tabs"/>
      <table width="100%" border="0" cellpadding="0" cellspacing="0">
        <tr>
          <td valign="top" width="230px">
            <div id="menu"/>
          </td>
          <td valign="top">
            <div id="main">
              <div id="breadcrumb"/>
              <div id="search"/>
              <div id="body"/>
            </div>
          </td>
        </tr>
        <tr>
          <td colspan="2" valign="top">
            <div id="footer">
              <div id="modified"/>
              <div id="publisher"/>
            </div>
          </td>
        </tr>
      </table>
      </div>
    </body>
  </html>
```

In the above example I have simply removed the html (in the template matching on the cmsbody element) from the page2xhtml.xsl stylesheet.

Once you have a template to use for your site, go ahead and save it into your publication. For me it made sense to create a templates directory in the resources directory of my publication and save my templates there. Note that I said templates, I found it very helpful to have a different template for each doc type that I use in my publication. Save your template as {yourpub}/resources/templates/xhtml.html (for an xhtml doc type).

Add your template to the aggregation in publication-sitemap.xmap

Open your publications publication-sitemap.xmap file for editing and search for cmsbody. Add your template to the aggregation like this:

```
      <!-- /lenyabody-{rendertype}/{publication-id}/{area}/{doctype}/{url} -->
      <map:match pattern="lenyabody-*/*/*/*/**">
        <map:aggregate element="cmsbody">
          <map:part src="cocoon://navigation/{2}/{3}/breadcrumb/{5}.xml"/>
          <map:part src="cocoon://navigation/{2}/{3}/tabs/{5}.xml"/>
          <map:part src="cocoon://navigation/{2}/{3}/menu/{5}.xml"/>
          <map:part src="cocoon://navigation/{2}/{3}/search/{5}.xml"/>
          <map:part src="cocoon:/lenya-document-{1}/{3}/{4}/{page-envelope:document-path}"/>

          <map:part src="resources/templates/{4}.html"/>
        </map:aggregate>
```

The {4}.html references your {doctype}.html template so make sure to have a template for EVERY doctype. If you only want to use one template you could do something like this: <map:part src="resources/templates/template.html"/>

Modify your page2xslt.xsl to utilize your template

You now need to remove the html template from your publications page2xslt.xsl stylesheet. Something like this below should work:

```
<?xml version="1.0" encoding="UTF-8" ?>

<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:xhtml="http://www.w3.org/1999/xhtml"
    xmlns:page="http://apache.org/cocoon/lenya/cms-page/1.0"
    xmlns:lenya="http://apache.org/cocoon/lenya/page-envelope/1.0"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:i18n="http://apache.org/cocoon/i18n/2.1"
    exclude-result-prefixes="page xhtml"
    >


<!-- {context-prefix}/{publication-id}/{area} -->
<xsl:param name="root"/>

<xsl:param name="document-id"/>

<!-- i.e. doctypes/xhtml-document -->
<xsl:param name="document-type"/>

<!-- The rquest url i.e. /lenya/doctypes/xhtml-document_en.html -->
<xsl:param name="url"/>

<xsl:param name="language"/>

<xsl:template match="cmsbody">
  <xsl:apply-templates select="xhtml:html"/>
</xsl:template>

<xsl:template match="xhtml:head">
  <xsl:copy>
    <xsl:apply-templates />
    <xsl:apply-templates select="//lenya:meta/dc:*" />
  </xsl:copy>
</xsl:template>

<xsl:template match="dc:*">
  <xsl:if test="text()">
    <xhtml:meta name="DC.{local-name(.)}" content="{text()}" />
  </xsl:if>
</xsl:template>

<xsl:template match="xhtml:title">
  <xsl:copy>
    <xsl:value-of select="//lenya:meta/dc:title"/>
  </xsl:copy>
```

```
</xsl:template>

<xsl:template match="xhtml:div[@id='menu']">
  <xsl:copy>
    <xsl:apply-templates select="@*"/>
    <xsl:apply-templates select="/cmsbody/xhtml:div[@id = 'menu']/*"/>
  </xsl:copy>
</xsl:template>

<xsl:template match="xhtml:td[@id='publication-title']">
  <xsl:copy>
    <xsl:apply-templates select="@*"/>
    <xsl:choose>
      <xsl:when test="$language = 'de'">
        Willkommen zur Default Publikation!
      </xsl:when>
      <xsl:otherwise>
        Welcome to the Default Publication!
      </xsl:otherwise>
    </xsl:choose>
  </xsl:copy>
</xsl:template>

<xsl:template match="xhtml:div[@id='tabs' and ancestor::xhtml:html]">
  <xsl:apply-templates select="/cmsbody/xhtml:div[@id = 'tabs']"/>
</xsl:template>

<xsl:template match="xhtml:div[@id='breadcrumb' and ancestor::xhtml:html]">
  <xsl:apply-templates select="/cmsbody/xhtml:div[@id = 'breadcrumb']"/>
</xsl:template>

<xsl:template match="xhtml:div[@id='search' and ancestor::xhtml:html]">
  <xsl:apply-templates select="/cmsbody/xhtml:div[@id = 'search']"/>
</xsl:template>

<xsl:template match="xhtml:div[@id='body' and ancestor::xhtml:html]">
  <xsl:copy>
    <xsl:apply-templates select="@*"/>
    <xsl:apply-templates select="/cmsbody/xhtml:div[@id = 'body']"/>
  </xsl:copy>
</xsl:template>

<xsl:template match="xhtml:div[@id='modified' and ancestor::xhtml:html]">
  <xsl:apply-templates select="/cmsbody/lenya:meta/dcterms:modified"/>
</xsl:template>

<xsl:template match="xhtml:div[@id='publisher' and ancestor::xhtml:html]">
  <xsl:apply-templates select="/cmsbody/lenya:meta/dc:publisher"/>
</xsl:template>

<xsl:template match="dcterms:modified">
  <xsl:variable name="date"><xsl:value-of select="."/></xsl:variable>
  <i18n:text>last_published</i18n:text>:
    <xsl:if test="$date!=''">
    <i18n:date-time src-pattern="yyyy-MM-dd HH:mm:ss" pattern="EEE, d MMM yyyy HH:mm:ss z" value="{$date}
"/>
  </xsl:if>
</xsl:template>

<xsl:template match="dc:publisher">
  <xsl:variable name="user"><xsl:value-of select="."/></xsl:variable>
  <xsl:variable name="user-id"><xsl:value-of select="substring-before($user,'|')"/></xsl:variable>
  <xsl:variable name="rest"><xsl:value-of select="substring-after($user,'|')"/></xsl:variable>
  <xsl:variable name="user-name"><xsl:value-of select="substring-before($rest,'|')"/></xsl:variable>
  <xsl:variable name="user-email"><xsl:value-of select="substring-after($rest,'|')"/></xsl:variable>

  <xsl:if test="$user != ''">
    <xsl:choose>
      <xsl:when test="$user-email != ''">
      / <a>
          <xsl:attribute name="href"><xsl:value-of select="concat('mailto:',$user-email)"/></xsl:attribut
```

```
e>
          <xsl:value-of select="$user-name"/>
        </a>
      </xsl:when>
      <xsl:otherwise>
       / <xsl:value-of select="$user-name"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:if>
</xsl:template>

<xsl:template match="@*|node()" priority="-1">
  <xsl:copy>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:copy>
</xsl:template>

</xsl:stylesheet>
```

Notice that you need to make templates (xsl:template - in the page2xhtml.xsl) that match on the container elements in your template (your xhtml template) and then apply the appropriate content using xslt.

Next steps

I would like to try using publication templating to make my templates be just simple documents in a parent publication and the css and images would just be assets of the document. This would give me versions of my templates, ease of changes, and all the other stuff Lenya provides.
Has anybody already done this, please continue this tutorial, feel free to correct me.