

Creating a Resource Type, Part 5: WYSIWYG Editing with BXE

Table of contents

1 Adding the Menu Item.....	2
2 Adding the webdavGET format.....	2
3 Adding BXE Support to the XSLT.....	2
4 Start Editing.....	3

In this section we make our person documents editable with the BXE WYSIWYG editor.

1. Adding the Menu Item

First, we add the BXE menu item to `$MODULE_HOME/config/menu.xsp`, right above the one-form editor item:

```
<item uc:usecase="bxedit" href="#"><i18n:text>With BXE</i18n:text></item>
```

2. Adding the webdavGET format

The BXE editor uses the WebDAV protocol to load the document source. To provide this functionality, we have to add the *webdavGET* format to our resource type declaration:

```
<component-instance name="person"
...
  <format name="webdavGET" uri="cocoon://modules/person/davget.xml"/>
</component-instance>
```

When you re-deploy the module, make sure that `cocoon.xconf` is properly updated - either by deleting the file from the web application, or by adding the new line manually.

The format is served by the following pipeline in `$MODULE_HOME/sitemap.xmap`:

```
<!-- webdav GET matcher -->
<map:match pattern="davget.xml">
  <map:act type="set-header">
    <map:parameter name="Last-Modified"
value="{date-iso8601-rfc822:{page-envelope:document-lastmodified}}"/>
    <map:generate src="lenya-document:{page-envelope:document-uuid}"/>
    <map:transform type="uuid2url">
      <map:parameter name="urls" value="absolute"/>
    </map:transform>
    <map:serialize type="xml"/>
  </map:act>
</map:match>
```

3. Adding BXE Support to the XSLT

Now we have to add specific `bxedit_xpath` attributes to the XHTML elements of our page so that BXE knows what can be edited. We update our stylesheet `$MODULE_HOME/xslt/foaf2xhtml.xsl` and add the attributes when the parameter `rendertype` equals `edit`:

```
<xsl:param name="rendertype"/>
...
<xsl:template name="bxeditAttribute">
  <xsl:param name="element"/>
  <xsl:if test="$rendertype = 'edit'">
    <xsl:attribute name="bxedit_xpath">
      /rdf:RDF/foaf:Person/foaf:<xsl:value-of select="$element"/>
    </xsl:attribute>
  </xsl:if>
</xsl:template>

<xsl:template match="foaf:Person">
  <h2>Person Details</h2>
  <table class="person">
    <tr>
```

```
<th>Title:</th>
<td>
  <xsl:call-template name="bxAttribute">
    <xsl:with-param name="element">title</xsl:with-param>
  </xsl:call-template>
  <xsl:value-of select="foaf:title"/>
</td>
</tr>
<tr>
<th>Given name:</th>
<td>
  <xsl:call-template name="bxAttribute">
    <xsl:with-param name="element">givenname</xsl:with-param>
  </xsl:call-template>
  <xsl:value-of select="foaf:givenname"/>
</td>
</tr>
<tr>
<th>Family name:</th>
<td>
  <xsl:call-template name="bxAttribute">
    <xsl:with-param name="element">family_name</xsl:with-param>
  </xsl:call-template>
  <xsl:value-of select="foaf:family_name"/>
</td>
</tr>
...
</xsl:template>
```

Now we're faced with a limitation of BXE: It is not possible to edit attributes in WYSIWYG mode. We'd have to change our XML structure to allow this. For this little tutorial, we'll just edit the element values - foaf:title, foaf:givenname, and foaf:family_name.

The rendertype parameter is passed to the stylesheet in \$MODULE_HOME/sitemap.xmap:

```
<map:transform src="fallback://lenya/modules/person/xslt/foaf2xhtml.xsl">
  <map:parameter name="rendertype" value="{request-param:rendertype}"/>
</map:transform>
```

4. Start Editing

Now you should be able to edit your person documents with BXE: