Kupu

Table of contents

1 Kupu as a sample for integrating editors in general....2
2 What integrating an editor is all about....2
3 The Kupu usecase pipeline in detail....3
3.1 step="open"....3
3.2 step="content"....3
3.3 step="save"....4

Kupu is a browser based WYSIWYG XHTML editor which is developed independently of Lenya by OSCOM (http://www.oscom.org/) and is available at http://kupu.oscom.org/ (http://kupu.oscom.org/) . Kupu can and actually is being used in other projects as well, that have nothing to do with Java, XML or the Apache software stack. Lenya is just prepared for the Kupu Editor to be plugged in, and ships with Kupu pre-installed.

1. Kupu as a sample for integrating editors in general

Understanding the Kupu integration in Lenya is helpful to understand how editors are integrated into Lenya at all. This knowledge is important if you are looking into changing the way Kupu is used inside Lenya or if you are thinking about taking any other editor and integrate it in Lenya.

2. What integrating an editor is all about

From the user's perspective using an editor in Lenya works like this:

1. Navigate to the item that is to be edited
2. Click on a link to start the editing process (mostly from a pull-down menu, but there are other options)
3. Edit
4. Save
5. Enjoy the changes

The link to start the editing is nohting but a GET request to an editing usecase. (See: [CreatePageWalkthrough (http://wiki.apache.org/lenya/CreatePageWalkthrough)] for some background on Usecases). The reply to that request is the page that contains the editor with the to be edited file loaded. As simple as that sounds, there is a number of things Lenya has to go through.

Between Step 2. and 3. (the user clicking on 'Edit' and the editor showing up) Lenya has to

1. Make sure the editor is installed at all
2. Determine the file to be edited
3. Make sure editing is allowed (i.e. the file is not checked out reserved by someone else)
4. Aggregate the content that is to be edited, the editor part of the page itself and whatever parameter the editor needs and serve it to the browser.

Between Step 3. and 4. (Edit and Save) Lenya has to

1. Make sure the file is still checked out so it can be edited
2. Extract the edited page content from the HTTP POST
3. Save the edited file
4. Check in the edited file
5. Trigger and workflow events on the document that belong to the "edit" event.

Note: Kupu allows the user to Save in between and continue editing. This is achieved by using HTTP POST operations. Thus Lenya needs to be prepared to receive multiple updates of the same page and

save it before the user might click on 'Exit and Save' so that Step 5 (Display the changed page) is finally triggered.

3. The Kupu usecase pipeline in detail

3.1. step="open"

```
        <map:match pattern="kupu" type="usecase">
          <map:match pattern="open" type="step">
            <map:match pattern="*/authoring/**.html">

              <!-- Check for Kupu-->
              <map:act type="resource-exists"
src="resources/kupu/common/kupueditor.js">

                <map:act type="reserved-checkout">
                  <map:generate type="serverpages" src="content/rc/{exception}.xsp">
                    <map:parameter name="user" value="{user}"/>
                    <map:parameter name="filename" value="{filename}"/>

                    <map:parameter name="date" value="{date}"/>
                    <map:parameter name="message" value="{message}"/>
                  </map:generate>
                  <map:transform src="xslt/rc/rco-exception.xsl"/>
                  <map:call resource="style-cms-page"/>
                </map:act>

                <map:generate src="resources/kupu/apache-lenya/kupumacros.html"/>
                <map:transform
src="resources/kupu/apache-lenya/kupumacros-xhtml.xsl">
                  <map:parameter name="contentfile"
value="{page-envelope:context-prefix}/{../1}/authoring/{../2}.html?lenya.usecase=kupu&amp;len
                  <map:parameter name="context" value="{request:contextPath}"/>
                  <map:parameter name="document-path"
value="{page-envelope:document-path}"/>

                  <map:parameter name="root"
value="{page-envelope:context-prefix}"/>
                  <map:parameter name="save-destination"
value="{page-envelope:context-prefix}/{../1}/authoring/{../2}.html?lenya.usecase=kupu&amp;len
                  <map:parameter name="reload-on-save" value="0"/> <!-- 1 means you
can see the saving -->
                  <map:parameter name="use-css" value="1"/>
                  <map:parameter name="redirect-to"
value="{page-envelope:document-name}.html"/>

                </map:transform>
                <map:serialize type="html"/>
              </map:act>
              <map:generate src="resources/misc/kupu/download.xhtml"/>
              <map:call resource="style-cms-page"/>
            </map:match>

          </map:match>
```

3.2. step="content"

```
          <map:match pattern="content" type="step">
            <map:match pattern="*/authoring/**.html">
              <map:generate
src="pubs/{1}/content/authoring/{page-envelope:document-path}"/>

              <map:transform
src="resources/kupu/apache-lenya/kupucontent-xhtml.xsl">
                <map:parameter name="css"
value="{page-envelope:context-prefix}/{1}/authoring/css/page.css"/>
              </map:transform>
              <map:transform src="xslt/util/strip_namespaces.xsl"/>
                <map:serialize type="xhtml"/>
            </map:match>

          </map:match>
```

3.3. step="save"

```
          <map:match pattern="save" type="step">
            <map:select type="request-method">
              <map:when test="PUT">

                <map:act type="reserved-checkout-test">
                  <map:generate type="serverpages"
src="../../content/rc/{exception}.xsp">
                    <map:parameter name="user" value="{user}"/>
                    <map:parameter name="filename" value="{filename}"/>
                    <map:parameter name="date" value="{date}"/>
                    <map:parameter name="message" value="{message}"/>

                  </map:generate>
                  <map:transform src="../../xslt/rc/rco-exception.xsl"/>
                  <map:call resource="style-cms-page"/>
                </map:act>

                <map:aggregate element="edit-envelope">
                  <map:part element="edited" src="cocoon:/kupu-stream"/>

                  <map:part element="original"
src="pubs/{page-envelope:publication-id}/content/authoring/{page-envelope:document-path}"/>
                </map:aggregate>

                <map:transform
src="resources/kupu/apache-lenya/kupusave-xhtml.xsl"/>

                <map:act type="reserved-checkin">
                  <map:generate src="content/rc/{exception}.xsp" type="serverpages">

                    <map:parameter name="user" value="{user}"/>
                    <map:parameter name="filename" value="{filename}"/>
                    <map:parameter name="checkType" value="{checkType}"/>
                    <map:parameter name="date" value="{date}"/>
                    <map:parameter name="message" value="{message}"/>
                  </map:generate>

                  <map:transform src="xslt/rc/rco-exception.xsl"/>
                  <map:serialize/>
                </map:act>
```

```
                <map:transform src="xslt/authoring/edit/addSourceTags.xsl">
                  <map:parameter name="source"
value="pubs/{page-envelope:publication-id}/content/authoring/{page-envelope:document-path}"/>
                </map:transform>

                <map:transform type="write-source"/>

                <!-- Trigger workflow with *edit* action -->
                <map:act type="workflow">
                  <map:parameter name="area" value="{page-envelope:area}"/>
                  <map:parameter name="document-id"
value="{page-envelope:document-id}"/>
                  <map:parameter name="language"
value="{page-envelope:document-language}"/>

                  <map:parameter name="event" value="edit"/>
                </map:act>

                <map:serialize type="xml" status-code="204"/>
              </map:when>
            </map:select>
          </map:match>
```