Integrating external data How-To

Table of contents

This article is about how to pull XML data from another web server and integrate it into a Lenya website. Your first decision is where Lenya enters your code. There are two good possibilities for your goals:

1. Interrupt before <map:match pattern="**.html">
2. Interrupt at getting the data.

1 First Solution

1. Interrupt before <map:match pattern="**.html"> Add a match and write everything needed to create your page. {1} = url before "/people/" {2} = url between "people/" and ".html"

```
            <map:match pattern="**/people/*.html">
                 <map:aggregate element="cmsbody">
                     <map:part src="cocoon://navigation/{page-envelope:publication-id}/{page-
envelope:area}/breadcrumb/index.xml"/>
                     <map:part src="cocoon://navigation/{page-envelope:publication-id}/{page-
envelope:area}/tabs/index.xml"/>
                     <map:part src="cocoon://navigation/{page-envelope:publication-id}/{page-
envelope:area}/menu/index.xml"/>
                     <map:part src="cocoon://navigation/{page-envelope:publication-id}/{page-
envelope:area}/search/index.xml"/>
                     <map:part src="cocoon:/people-{2}"/>
                 </map:aggregate>
                 <map:transform src="xslt/page2xhtml-people.xsl"/>
                 <map:serialize type="xml"/>
             </map:match>
```

(You should copy all the code in <map:match pattern="lenyabody-*/*/*/*/**">.

```
              <map:match pattern="people-**">
                 <map:generate src="http://www.xmlhack.com/cdf.cdf?{1}"/>
                 <map:transform src="xslt/xmlhack.xsl"/>
                 <map:serialize type="html"/>
             </map:match>
```

Notice I added the filename requested (without an extension) to the querystring of the remote request.

2 Second Solution

2. Interrupt at getting the data. 2.a Set a new doctype in "parameter-doctype.xmap". This code must be before <map:match pattern="*/**.html">:

```
<map:match pattern="**/people/*.html">
             <map:generate type="serverpages" src="../../config/parameters/default.xsp">
             <map:parameter name="value" value="people"/>
             </map:generate>
             <map:serialize type="xml"/>
             </map:match>
```

2.b Get the content from the remote source in "doctypes.xmap". This code must be before <map:match pattern="*/*/*/**.xml">: {1} = "view" {2} = area {3} = document=path with final ".xml" removed

```
             <map:match pattern="*/*/people/**.xml">
                 <map:generate src="http://www.xmlhack.com/cdf.cdf?{3}"/>
                 <map:transform src="xslt/people2xhtml.xsl">
                     <map:parameter name="rendertype" value="{1}"/>
```

```
                      <map:parameter name="nodeid" value="{page-envelope:document-node-id}"/>
                      <map:parameter name="language" value="{page-envelope:document-language}"/>
                  </map:transform>
                  <map:serialize type="xml"/>
              </map:match>
```

Notice I added the filename requested (without an extension) to the querystring of the remote request.

2.c Use these filenames for your transformation: *xslt/people2xhtml.xsl* (Use whatever is specified in 2.b, but this is the standard naming convention.) *xslt/page2xhtml-people.xsl* I recommend the second option. It is much less code (less chance of bugs), and takes advantage of Lenya's standards.