Creating a Resource Type, Part 1: Prerequisites and Declaration

Table of contents

This section explains how to create a new resource type. For more information on resource types, refer to the resource types reference (../../../../docs/2_0_x/reference/resource-types.html) . For the sake of simplicity, we'll test the resource type with the default publication.

Our resource type will support storing details (name, address, etc.) of people. We'll call it *person*. Since we're too lazy (and too smart) to invent our own XML schema, we'll just use the one provided by the FoaF (http://www.foaf-project.org/) project.

You can checkout the source code of this example from the Subversion repository at the URL `http://svn.apache.org/repos/asf/lenya/sandbox/modules/person`.

1. Prerequisites

We'll use the following directory layout:

```
$HOME/
  apache/
    lenya-2.0/             The Lenya installation directory, we'll call it $LENYA_HOME.
  src/
    lenya/                 The home directory of your Lenya-related sources.
      modules/             Our modules.
        person/            Our "person" resource type module ($MODULE_HOME).
```

2. Adding a Module

We'll add a module which will contain the resource type. This makes it self-contained, and it will be very easy to re-use our resource type in different publications. Create the *person* directory (see section *Prerequisites*) with the following directory layout:

```
person/
  config/                        Configuration of the module.
    menu.xsp                     The resource type menu items.
    module.xml                   Module descriptor.
    cocoon-xconf/                Patch files for cocoon.xconf.
      resourcetype-person.xconf  The declaration of our resource type.
  resources/
    i18n/                        The internationalization catalogues.
    icons/
      person.gif                 The icon for the sitetree.
  samples/                       Samples to create person documents.
    foaf.xml                     The default sample document.
  xslt/
    foaf2xhtml.xml               Transform FoaF documents to XHTML.
  menus.xmap                     The sitemap generating the menu.
```

Each module needs a module descriptor file `$MODULE_HOME/config/module.xml`. In our case it looks like this (replace `org.yourproject` with your own package name):

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="http://apache.org/lenya/module/1.0">
  <id>org.yourproject.lenya.modules.person</id>
  <package>org.yourproject.lenya.modules</package>
  <version>0.1-dev</version>
  <name>person</name>
  <lenya-version>@lenya.version@</lenya-version>
  <description>Resource type to store person details</description>
</module>
```

Now we have to let Lenya know that we've added a new module. Edit the file

`$LENYA_HOME/local.build.properties` and add the path to your `modules` directory (the parent of `$MODULE_HOME`) at the end of the `modules.root.dirs` declaration:

```
modules.root.dirs=...:/home/john/src/lenya/modules
```

Finally, we'll have to declare the module in the publications which use it (in our case, the default publication). Edit `$PUB_HOME/config/publication.xml` and add the entry

```
<module name="person"/>
```

to the `<modules/>` section. Additionally, you can assign a workflow to the resource type in the `<resource-types>` section of `publication.xml`:

```
<resource-type name="person" workflow="fallback://config/workflow/workflow.xml"/>
```

3. Declaring the Resource Type

To let Lenya know that a new resource type exists, we'll add the resource type declaration file `resourcetype-person.xconf`. This is a patch for the `cocoon.xconf` file and therefore located in `$MODULE_HOME/config/cocoon-xconf`.

```
<?xml version="1.0"?>
<xconf xpath="/cocoon/resource-types"
       unless="/cocoon/resource-types/component-instance[@name = 'person']">

  <component-instance name="person" logger="lenya.resourcetypes"
    class="org.apache.lenya.cms.publication.ResourceTypeImpl">

    <schema
       namespace="http://relaxng.org/ns/structure/0.9"
       uri="fallback://lenya/modules/person/resources/schemas/foaf.rng"
    />

    <!-- Default time cache time in seconds for this resource type -->
    <expires seconds="3600" />

    <sample
       name="Basic FOAF sample"
       mime-type="application/rdf+xml"
       uri="fallback://lenya/modules/person/samples/foaf.xml"
    />

    <format name="xhtml" uri="cocoon://modules/person/xhtml.xml"/>
    <format name="xhtml-include" uri="cocoon://modules/person/xhtml-include.xml"/>
    <format name="icon" uri="cocoon://modules/person/icon"/>

  </component-instance>

</xconf>
```

According to this declaration, the Lenya web application creates a new `ResourceTypeImpl` service on startup which makes the resource type details available to other objects.

In the next section (../../../../docs/2_0_x/tutorials/resourcetype/part2.html) we'll setup the creation of person documents.

4. Adding I18n Messages for the Resource Type Name

Now we'll provide i18n messages for each language to show the resource type name in a

human-readable form. There is a convention that the message key *resourceType-{name}* is used.

Create the file `$MODULE_HOME/resources/i18n/cmsui.xml` with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<catalogue xml:lang="en" xmlns:xhtml="http://www.w3.org/1999/xhtml">
  <message key="resourceType-person">Person</message>
</catalogue>
```

For other languages, use the language code as suffix (`cmsui_de.xhml` etc.) and don't forget to set the *xml:lang* attribute of the `<catalogue>` element.