

Modules

Table of contents

1 Introduction.....	2
2 Implementing a module.....	2
3 Implementing Usecases in Modules.....	2
3.1 Sitemaps for Module Usecases.....	3

1. Introduction

Modules are packages providing a certain set of resources or functionality.

Examples:

- a resource type (e.g., docbook module)
- a repository implementation (e.g., jdbc module)
- a collection of XSLTs (e.g., content2svg module)

Some modules are included in the Lenya distribution:

- **search** - search functionality
- **sitetree** - manage documents in a tree-like structure
- **jcr** - store content in a [JCR](http://www.jcp.org/en/jsr/detail?id=170) (<http://www.jcp.org/en/jsr/detail?id=170>) repository
- **xhtml** - XHTML-based resource type
- **links** - resource type to manage link lists
- **lenyadoc** - adds the `lenyadoc://` protocol

2. Implementing a module

All resources of a module are located in a single directory. The following is an example directory structure, all files are optional and depend on the nature of the module.

```
mymodule/
  config/
    components.xconf      configuration files
    patch for cocoon.xconf
  usecases/
    mymodule.jx           usecase view
  java/
    src/                  Java source files
    lib/                  Java libraries
  xslt/                   XSLT stylesheets
  sitemap.xmap            main module sitemap
```

To add a module to your Lenya installation, declare it in `local.build.properties`:

```
modules.root.dirs=src/modules:src/webapp/lenya/pubs/default/modules:/home/john/modules/mymodule
```

When the module is deployed, the following steps are executed:

- the module files are copied to `context://lenya/modules`
- Java sources are compiled, libraries are installed
- `cocoon.xconf` is patched

3. Implementing Usecases in Modules

To declare usecases, add a patch file for `cocoon.xconf`, for instance `module/config/usecases.xconf`:

```
<xconf xpath="/cocoon/usecases"
  unless="/cocoon/usecases/component-instance[@name = 'mymodule.edit']">
  <component-instance name="mymodule.edit"
    logger="lenya.usecases.edit
    class="org.myproject.lenya.mymodule.usecases.Edit">
    <view template="modules/mymodule/usecases/edit.jx" menu="false"/>
  </component-instance>
</xconf>
```

As you can see in the view declaration, the JX templates are typically stored in the `mymodule/usecases` directory. The Java source files go into the `mymodule/java/src` directory, they are compiled automatically by the Lenya build process.

3.1. Sitemaps for Module Usecases

Module-specific usecase sitemaps must be located at

```
<module>/usecases/<usecase-name>/usecase.xmap
```

To make a call into a usecase sitemap, use a request of the following form:

```
cocoon://core/modules/<module>/usecases/<usecase-name>/<request>
```

For example, imagine you implement a newsletter module, containing a send usecase. The usecase sitemap is located at `mymodules/newsletter/usecases/send/usecase.xmap`. To request the latest newsletter and display it on the confirmation screen using the CInclude approach (for more information, see documentation about the usecase framework), you could for instance use the URI

```
cocoon://core/modules/newsletter/usecases/send/latestNewsletter.xml:
```

```
<page:page
  xmlns:jx="http://apache.org/cocoon/templates/jx/1.0"
  xmlns:page="http://apache.org/cocoon/lenya/cms-page/1.0"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:il8n="http://apache.org/cocoon/il8n/2.1"
  xmlns:cinclude="http://apache.org/cocoon/include/1.0"
>

  <page:body>
    <h1><il8n:text>Send Newsletter</il8n:text></h1>
    <form>
      ...
      <jx:import uri="templates/messages.jx"/>

      <cinclude:includexml>
<cinclude:src>cocoon://core/modules/newsletter/usecases/send/latestNewsletter.xml</cinclude:src>
      </cinclude:includexml>

      <input name="submit" type="submit" value="Send Newsletter"/>
    </form>
  </page:body>
</page:page>
```