Policies and Policy Managers

Table of contents

1 Policies

A `Policy` assigns `Role`s to `Accreditable`s.

There is a common policy definition XML schema which is handled by the `PolicyBuilder`. It can be used together with the `FilePolicyManager` and the `SitemapPolicyManager`.

Here is an example of a policy definition:

```
<policy xmlns="http://apache.org/cocoon/lenya/ac/1.0">

  <user id="lenya">
    <role id="editor"/>
    <role id="reviewer"/>
  </group>

  <group id="reviewer">
    <role id="reviewer"/>
  </group>

  <ip-range id="localhost">
    <role id="editor"/>
  </ip-range>

  <world>
    <role id="visitor"/>
  </world>

</policy>
```

2 Policy Managers

A *PolicyManager* is used to resolve the policy for a certain URL. Lenya ships with the following *PolicyManager*s:

3 Inheriting Policy Manager

This is an abstract base class. It merges the policies of all steps in the URL. For each URL, a *url policy* and a *subtree policy* can be defined. The *InheritingPolicyManager* adds the credentials of

- the subtree policies for all parent directories of the requested page,
- the subtree policy of the requested page, and
- the url policy of the requested page.

For instance, if the URL is `/lenya/news/index.html`, the following policies are merged:

- subtree policy of `/`
- subtree policy of `/lenya`
- subtree policy of `/lenya/news`
- subtree policy of `/lenya/news/index.html`
- url policy of `/lenya/news/index.html`

4 File Policy Manager

The *FilePolicyManager* is an *InheritingPolicyManager*. The policies are defined by policy files that are arranged as a directory tree that reflects the URI space, e.g.:

```
/subtree-policy.acml
/lenya/subtree-policy.acml
/lenya/news/index.html/subtree-policy.acml
/lenya/news/index.html/url-policy.acml
```

If a certain policy file does not exist (like /lenya/news in the above example), an empty policy is used instead.

The *FilePolicyManager* needs a `directory` parameter which contains a URL pointing to the policies directory:

```
<policy-manager type="file">
  <parameter name="directory"
      value="context:///lenya/pubs/mypub/config/ac/policies"/>
</policy-manager>
```

5 Document Policy Manager Wrapper

This *InheritingPolicyManager* subclass is used together with another *InheritingPolicyManager*. It is able to apply a single policy to all versions of a document (languages, print version, ...). E. g., if you define

- `/foo/bar/subtree-policy.xml`

and you use the *DefaultDocumentBuilder*, this policy is applied to the URLs

- `/foo/bar.html`
- `/foo/bar_de.html`
- `/foo/bar_en.print.html`
- ...

To configure the *DefaultDocumentBuilder*, just put the declaration of the wrapped *PolicyManager* inside the *DefaultDocumentBuilder* declaration:

```
<policy-manager type="document">
  <policy-manager type="file">
    <parameter name="directory"
        value="context:///lenya/pubs/mypub/config/ac/policies"/>
  </policy-manager>
</policy-manager>
```

6 Sitemap Policy Manager

The *SitemapPolicyManager* uses the policy sitemap to resolve the policy for a certain URL. For this purpose it sends a request of the form

```
cocoon://{publication-id}/policies{url}.acml

Example:
cocoon://mypub/policies/authoring/foo/bar_de.html.acml
```

which is processed by `global-sitemap.xmap` and forwarded to `lenya/pubs/{publication-id}/policies-sitemap.xmap`. The request is supposed to return a valid policy XML document.

The configuration of the *SitemapPolicyManager* is very simple:

```
<policy-manager type="sitemap"/>
```