Access Controller Resolvers

Table of contents

1. Introduction

In Lenya, the *AccessController* is chosen dynamically, depending on the requested URL. An *AccessControllerResolver* resolves the appropriate *AccessController* for a URL.

2. Defining the Roles in lenya.roles

To obtain an *AccessControllerResolver*, the *AccessControllerResolverSelector* is used. A component which wants to obtain an *AccessController* asks the selector for the *ComposableAccessControllerResolver*. The *ComposableAccessControllerResolver* itself uses the selector to obtain its child *AccessControllerResolver*s.

```
<role name="org.apache.lenya.ac.AccessControllerResolverSelector"
    shorthand="access-controller-resolvers"
default-class="org.apache.avalon.excalibur.component.ExcaliburComponentSelector">
  <hint shorthand="configurable"
      class="org.apache.lenya.ac.impl.ConfigurableAccessControllerResolver"/>
  <hint shorthand="publication"
      class="org.apache.lenya.cms.ac.PublicationAccessControllerResolver"/>
  <hint shorthand="composable"
      class="org.apache.lenya.ac.impl.ComposableAccessControllerResolver"/>
</role>
```

3. Declaring the Access Controller Resolvers in cocoon.xconf

```
<access-controller-resolvers>
  <component-instance logger="lenya.ac.accesscontrollerresolver.publication"
    class="org.apache.lenya.cms.ac.PublicationAccessControllerResolver"
    name="publication">
  </component-instance>
  <component-instance logger="lenya.ac.accesscontrollerresolver.global"
    class="org.apache.lenya.ac.impl.ConfigurableAccessControllerResolver"
    name="global">
    <access-controller type="global"/>
  </component-instance>
  <component-instance logger="lenya.ac.accesscontrollerresolver.composable"
    class="org.apache.lenya.ac.impl.ComposableAccessControllerResolver"
    name="composable">
    <resolver type="publication"/>
    <resolver type="global"/>
  </component-instance>
</access-controller-resolvers>
```

4. Publication Access Controller Resolver

The *PublicationAccessControllerResolver* looks for a `config/ac.xconf` file inside the publication. If you want to use multiple *AccessController*s within your Lenya installation, just declare them in the `cocoon-xconf.xsl` file and choose the type in the `config/ac.xconf` file. The `type` attribute selects an *AccessController* from the definitions in `cocoon.xconf`.

You have to configure the complete *AccessController* in this file. For instance, if you want to use a *BypassableAccessController* together with a certain set of components, you declare it as follows:

```
<?xml version="1.0"?>
<access-controller type="bypassable">
```

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

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

  <authorizer type="policy"/>

  <authorizer type="usecase">
    <parameter name="configuration"
        value="context:///lenya/pubs/default/config/ac/usecase-policies.xml"/>
  </authorizer>

  <authorizer type="workflow"/>

</access-controller>
```

5. Configurable Access Controller Resolver

The *ConfigurableAccessControllerResolver* can be configured with an *AccessController* directly inside `cocoon.xconf`:

```
<component-instance logger="lenya.ac.accesscontrollerresolver"
    class="org.apache.lenya.ac.impl.ConfigurableAccessControllerResolver"
    name="global">
  <access-controller type="global"/>
</component-instance>
```

6. Composable Access Controller Resolver

The *ComposableAccessControllerResolver* is configured with a list of *AccessControllerResolver*s. Each one of these resolvers is invoked until one is successful. If no resolver finds an *AccessController*, the *ComposableAccessControllerResolver* returns `null`.

```
<component-instance logger="lenya.ac.accesscontrollerresolver"
    class="org.apache.lenya.ac.impl.ComposableAccessControllerResolver"
    name="composable">
  <resolver type="publication"/>
  <resolver type="global"/>
</component-instance>
```