

Configuration guide

Esup Lecture Portlet - Configuration Guide



Auteur : Bourges Raymond - Bouteille Gwénaëlle - Repain Vincent ([University of Rennes 1](#))

- [Esup Lecture Portlet - Configuration Guide](#)
- [Configuration](#)
 - [Technical configuration](#)
 - [Content configuration](#)
 - [esup-lecture.xml](#)
 - [Element channelConfig](#)
 - [Element guestUser](#)
 - [Element ttl](#)
 - [Element context](#)
 - [Element categoryProfile](#)
 - [Element visibility](#)
 - [<category>.xml](#)
 - [element category](#)
 - [element sourceProfile](#)
 - [mappings.xml](#)
 - [portlet.xml](#)
 - [auth.xml](#)

Configuration

Technical configuration

Adapt to your environment these configuration files:

- **logging/log4j.properties** for logging configuration
- **esup-lecture.xml** for global configuration

Content configuration

To adapt to your environment, edit these configuration files:

- **esup-lecture.xml**: main configuration file. Contexts and CategoryProfiles definition. It also deals with **<category>.xml**: remote xml file referenced by CategoryProfiles.
- **mappings.xml**: declarations about xslt transformation for interface display
- **portlet.xml**: it needs declaration of portal user attributes used by Lecture Portlet
- **auth.xml**: auth configuration



XML Elements or attributes not explained here (but in dtd) implements features that are not yet supported.

esup-lecture.xml

This file contains some technical parameters (guestUser, ttl) and describes the contents displayed (contexts, category profiles). Here is the structure of this file (for more information, look at dtd **esup-lecture.dtd**):

Element channelConfig

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE channelConfig SYSTEM "lecture-config.dtd" >
<channelConfig>

  <!-- guestUser definition -->
  <guestUser> ... </guestUser>

  <!-- ttl definition -->
  <ttl> ... </ttl>

  <!-- contexts definition -->
  <context> ... </context>
  <context> ... </context>
  ...

  <!-- category profile definition -->
  <categoryProfile> ... </categoryProfile>
  <categoryProfile> ... </categoryProfile>
  ...

</channelConfig>
```

channelConfig is the root element, it is of course mandatory and all others elements must be described under it.

Element guestUser

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE channelConfig SYSTEM "lecture-config.dtd" >
<channelConfig>
  <guestUser>guest</guestUser>
```

guestUser is an optional property. Default value is "guest".

If the current connected user name equals **guestUser** property, then all controls used for personalisation (change tree size buttons, mark an item as read button, edit button, etc.) are hidden.

This is used in Portlet mode when esup-lecture is used in a portal unauthenticated view.

Element ttl

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE channelConfig SYSTEM "lecture-config.dtd" >
<channelConfig>
  <ttl>0</ttl>
```

ttl is an optional property. Default value is 0 (zero). If not used, the configTtl property of bean channel (see properties/domain/domain.xml) is used instead.

It tunes the time in seconds the two configuration files (esup-lecture.xml and mappings.xml) will stay in cache, that is, the time between two reloadings of these files. This allow to change the configuration files without reloading the portlet. A value of zero means no reloading.

Element context

```
<context
  name = "Démonstration Esup-Lecture"
  id = "default">
  <description>Context de démo</description>
  <refCategoryProfile refId="demo1"/>
  <refCategoryProfile refId="demo2"/>
  ...
</context>
```

You can use a specific context by configuring your portlet with a portlet preference. This portlet preference must have a name "context" and a value equals to the id you want for this portlet. If you don't define any context preference for your portlet then context with id "default" is used. With this mechanism you can define many channels with many contexts definitions in your portal environment with just one instance of the portlet.



Before uPortal 2.5.4, because of a bug (<http://www.ja-sig.org/issues/browse/UP-1040>) you can't use this feature.

A context has the following attributes :

- *id*
- *name* (displayed on interface)
- *description* (displayed on interface)
- *treeVisible* (yes | no) : If "yes", users can see the left panel (tree of categories and sources). Yes by default

A context may have also the following elements :

- *refCategoryProfile* references a category profile declared in this context (using category profile id). You can declare as many category profiles as you want in a context. Each category profile must be defined in an element *categoryProfile*. A same *categoryProfile* can be declared in several contexts. *refCategoryProfile* has the following attribute :
 - *refId* : Identify the category profile (attribute *id* of element *categoryProfile*, see below)
- *categoryProfilesUrl* is an alternative way to declare category profiles, which are in this case defined externally. It is used in coordination with applications who are able to show urls containing categoryProfiles definitions. *categoryProfilesUrl* has the following attributes :
 - *url* : The url containing categoryprofiles definitions. This url must be a xml file with a root element containing *categoryprofile* elements (see below), like this :

```
<?xml version="1.0" encoding="UTF-8"?>

<categoryProfilesUrl>
  <categoryProfile>...</categoryProfile>
  <categoryProfile>...</categoryProfile>
  ...
</categoryProfilesUrl>
```

- *timeout* (milliseconds) : time trying to get the *categoryProfilesUrl*. If unsuccessful, a new attempt will be made after the time indicated in *defaultTtl* property (in seconds) of bean channel (see properties/domain/domain.xml)
- *idPrefix* : prefix added to categoryProfiles Ids, to avoid conflicts with other categoryProfiles definitions.

Element categoryProfile

```
<categoryProfile  name="Categorie de démo"
  id="demo1"
  urlCategory="http://partages.univ-rennes1.fr/files/partages/Services/CRI/SI/conf_lecture_gwe_ray/demo1.xml"
  trustCategory="no"
  access="public"
  ttl = "3600"
  timeout = "3000">
  <visibility> ... </visibility>
</categoryProfile>
```

A category profile have the following attributes :

- *id*,
- *name*,
- *urlCategory* : url to get back a xml category file on a remote server
- *trustCategory* (yes | no) : If it is "yes", visibility rights used are those of remote category and sources. If it is "no", visibility rights on category and sources used are those of this category profile, defined in *visibility* element.
- *access* (public | cas) : access of the category is *public* or *cas* because it needs CAS proxy ticket for authentication
- *ttl* (seconds) : time to live of the remote category and its sources
- *timeout* (milliseconds) : time trying to get the category
- *visibility* : define group visibility for category referenced by this category profile. It is used only if *trustCategory* is set to "no"
- *userCanMarkRead* (yes | no) : If "yes", users can mark items of the category as read or not read. Yes by default"

Element visibility

```

<visibility>
  <allowed/>
  <autoSubscribed/>
  <obliged>
    <group name="local.0" />
    <group .../>
    ...
    <regular attribute="sn" value="user" />
    <regular .../>
    ...
    <regex attribute="isMemberOf" pattern="[ a-zA-Z0-9:]*:Eleves" />

    <regex .../>
    ...
  </obliged>
</visibility>

```

In this element, you define 3 groups of visibility;

- *allowed* : users are not automatically subscribed to this category, but are allowed to subscribe.
- *autoSubscribed* : users are automatically subscribed to this category and are allowed to unsubscribe
- *obliged* : users are automatically subscribe to this category and can't unsubscribe

A user is in a visibility group by two ways :

- *group* : user is in the portal group referenced by attribute *name*, in the example : user in in group "local.0" - see "portlet.xml" section.
- *regular* : user check regular, in the example : user value of portal attribute "sn" is "user" - see "portlet.xml" section.
- *regex* : user check a java regular expression from a portal user attribute, in the example user value of portal attribute "isMemberOf" is matching the pattern corresponding of all members of a group when the name contains these characters "[a-zA-Z0-9:]" 0 or n times and when the name finish by ":Eleves" - see "portlet.xml" section.

<category>.xml

described by dtd **category.dtd**, provided by remote server, requested by *urlCategory* of esup-lecture.xml

element category

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE category SYSTEM "category.dtd">
<category name="Différents RSS">
  <description>Très belle description !</description>
  <sourceProfiles>
    <sourceProfile> ... </sourceProfiles>
    <sourceProfile> ... </sourceProfiles>
    ...
  </sourceProfiles>
  <visibility> ... </visibility>
</category>

```

A category has a *name* and a *description* (displayed on interface), and

- a list of *source profiles*
- *visibility* : define group visibility for this category. It is used only if *trustCategory* attribute of referencing *categoryProfile* is set to "yes" else visibility of category profile is used - optional -

element sourceProfile

```

<sourceProfile
  id="un" access="public" name="Incidents techniques Rennes 1"
  specificUserContent="no" url="http://info.cri.univ-rennes1.fr/rss/rss.php">
  <visibility> ... </visibility>

</sourceProfile>

```

A source profile has an *id*, a *name* (displayed on interface) and :

- *access* (public | cas) : access of the source is *public* or *cas* because it needs CAS proxy ticket for authentication

- *specificUserContent* (yes | no) : if it is "yes", source content is specific to user. If it is "no", source content is the same for every users (If your configure with "yes" then application assumes that content can be deferent for each user (may be because of specific content due to profiling according to authentication). In this case application doesn't use any cache for the source. So be careful before use "yes" for this property.)
- *url* : url to get xml stream of the source
- *timeout* (milliseconds) : Time trying to get the source. Parent category timeout is used is it is not defined here
- *visibility* : define group visibility for source refered by this source profile. It is used only if *trustCategory* attribute of referencing *categoryProfile* is set to "yes" else visibility category profile is used - optional -



Be carefull to manage unique id for every sources profiles defined in categories : application does not yet manage it.

mappings.xml

mappings.xml: it describes a list of mappings used to parse xml stream of a source in a html content (see **mappings.dtd**) :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mappings SYSTEM "mappings.dtd" >
<mappings>
  <mapping> ... </mapping>
  <mapping> ... </mapping>
  ...
</mappings>
```

A source is composed of a list of items that will be parsed to be displayed in web page. A mapping is used to define a *xsltFile* to apply on items selected by *itemXPath*. Sometimes, to define *itemXPath*, namespaces definition are required.

Here is an example of mapping:

```
<mapping
  sourceURL="http://info.cri.univ-rennes1.fr/rss/rss.php"
  xsltFile="http://partages.univ-rennes1.fr/files/partages/Services/CRI/SI/conf_lecture_gwe_ray/styleSheet01.xsl"
  itemXPath="/rdf:RDF/default:item">
  <XPathNameSpace prefix="rdf" uri="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <XPathNameSpace prefix="default" uri="http://purl.org/rss/1.0/" />
</mapping>
```

Explanations:

- *xsltFile* : url of the xslt file used to parse an item
- *mobileXsltFile* : url of the xslt file used to parse an item in mobile mode - optionnal. If not present *xsltFile* is also used in mobile mode.
- *itemXPath* : xpath expression to locate an item in the xml stream source
- element *XPathNameSpace* : used by *itemXPath* definition - optionnal
- In order to define which mapping is applied to a particular source some attributes are used (by priority order):
 - *sourceURL* : url of the source, key entry of this mapping (example: sourceURL="http://info.cri.univ-rennes1.fr/rss/rss.php")
 - *dtd* : dtd of the source
 - *xmlType* : xmlType of the source
 - *xmlns* : xml namespace of the source
 - *rootElement* : rootElement of the source

Priority to find xslt informations on key entry of a mapping are : sourceURL, DTD, xmlType, xmlns and finally rootElement.

portlet.xml

All portal user attributes used by portlet must be declared in the **webapp/WEB-INF/portlet.xml**, here is an example:

```
<user-attribute>
  <description>the username of the portal user</description>
  <name>username</name>
</user-attribute>
<user-attribute>
  <description>the displayName of the portal user</description>
  <name>displayName</name>
</user-attribute>
<user-attribute>
  <description>the sn of the portal user</description>
  <name>sn</name>
</user-attribute>
```

auth.xmlm

This file (**properties/auth.xml**) is used to define auth mechanism. For example when using CAS in servlet mode or a portal in portlet mode:

```
<bean id="authenticationService"
  class="org.esupportail.commons.services.authentication.PortalOrCasFilterAuthenticator">
  <description>The name of the Portal attribute that holds the uid of users, as set in portlet.xml.<
/description>
  <property name="uidPortalAttribute" value="uid" />
</bean>
```