

Indexation XSLT

Nous avons vu que pour indexer un document dans Solr, il faut utiliser le format XML. Nous savons comment transformer des documents XML d'un format à un autre en utilisant des XSLT - via un éditeur XML (Oxygene, Netbeans, Liguix, etc). Solr permet de le faire aussi, via une url d'indexation spécifique.

Pour indexer des documents dans Solr, l'URL standard est:

```
http://localhost:8983/solr/update
```

L'indexation en passant par une XSLT se fait via cet URL:

```
http://localhost:8983/solr/update/xslt
```

En passant dans le paramètre 'tr' le nom de la XSLT à utiliser, qui doit se trouver dans le répertoire:

```
<path_to_solr>/solr/conf/xslt
```

Exemple:

```
http://localhost:8983/solr/update/xslt?tr=comem-to-solr.xsl
```

Utilisation avec post.jar

L'outil d'indexation post.jar utilise l'URL d'indexation par défaut de Solr. Du coup, il faut lui spécifier un URL différent pour pouvoir utiliser la méthode XSLT. Cela se fait en utilisant le paramètre -Durl=<url>. Exemple:

```
java -Durl=http://localhost:8983/solr/update/xslt?tr=comem-to-solr.xsl -jar post.jar <fichier_xml>
```

Pour l'exécution de cette commande, on part du principe qu'on se trouve actuellement dans le dossier de Solr, si ce n'est pas le cas, il faut s'y rendre préalablement ou spécifier le chemin complet du fichier post.jar.

Test de la XSLT

Au cas où la XSLT n'est pas fonctionnelle, Solr ne va pas être très verbeux. Il indiquera seulement dans le log une erreur du genre:

```
SimplePostTool: FATAL: Solr returned an error #500 Unable to initialize Templates 'comem-to-solr.xsl'
```

Ce qui ne nous permet pas de savoir où se trouve le problème.

Il faut donc utiliser un autre outil pour déboguer la XSLT. Par exemple la commande 'xsltproc', disponible sur la plupart des systèmes UNIX, qui permet d'appliquer une XSLT sur un fichier XML quelconque:

```
xsltproc <path_to_xsl> <path_to_xml>
```

Exemple:

```
xsltproc solr/conf/xslt/comem-to-solr.xsl delivrables/text/un-texte.xml
```

Elle va retourner sur la ligne de commande le fichier xml résultant ou les erreurs eventuelles.

Pour un affichage plus lisible, ajouter dans la XSLT, l'instruction XSL suivante:

```
<xsl:output method="xml" indent="yes"/>
```

Durée du cache XSLT de Solr

Par défaut, Solr est configuré pour garder en mémoire toutes les XSLT durant 60 secondes. Cela signifie que si des modifications sont effectuées sur l'une ou l'autre des XSLT présentes dans le répertoire solr/conf/xslt, il faudra attendre au maximum 60 secondes avant que les changements soient pris en compte par Solr.

Pour éviter ce comportement (qui peut être gênant lorsqu'on fait des tests), il faut changer la configuration du requestHandler correspondant à l'url /solr/update/xslt (utilisé pour indexer les fichiers XML via une XSLT).

Rechercher dans le fichier /solr/conf/solrconfig.xml, le requestHandler '/update/xslt' et le modifier ainsi:

```
<requestHandler name="/update/xslt"
  startup="lazy"
  class="solr.XsltUpdateRequestHandler">
  <int name="xsltCacheLifetimeSeconds">0</int>
</requestHandler>
```

Puis redémarrer Solr pour qu'il prenne en compte le changement.