Space Details

Key:	ORIOAIindexing
Name:	ORI-OAI-indexing
Description:	
Creator (Creation Date):	ycolmant@univ-valenciennes.fr (Jul 04, 2008)
Last Modifier (Mod. Date):	ycolmant@univ-valenciennes.fr (Aug 29, 2008)

Available Pages

- Version 1.1
  - Spécifications
    - Indexing dans Ori-Oai
    - Lucene et applications attachées
    - OSCache
  - Installation
    - Introduction
    - Configuration pour un deploiement rapide
    - Premiers tests
    - Personnalisation de la configuration
  - Utilisation
    - Connexion au Web Service
    - Methodes publiques
    - La classe SearchResults
    - Administration
    - Contraintes d'utilisation

Version 1.1

This page last changed on Nov 25, 2008 by ycolmant@univ-valenciennes.fr.

ORI-OAI-indexing : Indexation des ressources

Module obligatoire quelque soit la configuration choisie

Voir l'architecture du système

Description

Une fois le dépôt de ressources et la saisie de métadonnées validés, ces dernières sont indexées par le module ORI-OAI-indexing. Ce module a pour rôle l'indexation des fiches de métadonnées ainsi que des documents associés.

Pour cela, il utilise le moteur d'indexation Lucene. Celui-ci permet l'indexation de différentes sources offrant une recherche puissante et rapide en se reposant sur différents analyseurs. L'analyseur de la langue française permettra notamment la gestion des verbes conjugués, des pluriels ou encore des accents et caractères spéciaux. Un système de pondération permet aussi de rendre une métadonnée plus pertinente qu'une autre. Par exemple, on préfèrera retrouver en premier les documents dont l'élément recherché se trouve dans le titre plutôt que dans la description.

Lius est un framework d'indexation basé sur le projet Lucene. Il permet une indexation de différents formats de fichiers comme XML, PDF, OpenOffice, ZIP, MP3, etc. Il est utilisé dans notre projet pour offrir une configuration avancée des champs à indexer dans les différents formats de fiches de métadonnées XML et, par la suite, pour indexer les documents associés en plein texte.

En plus de l'aspect indexation, ORI-OAI-indexing offre un service de recherche de documents via Web service en se reposant sur la syntaxe des requêtes Lucene. Il est utilisé par différents composants dans le système.

Voir la documentation technique

Spécifications

This page last changed on Nov 25, 2008 by ycolmant@univ-valenciennes.fr.

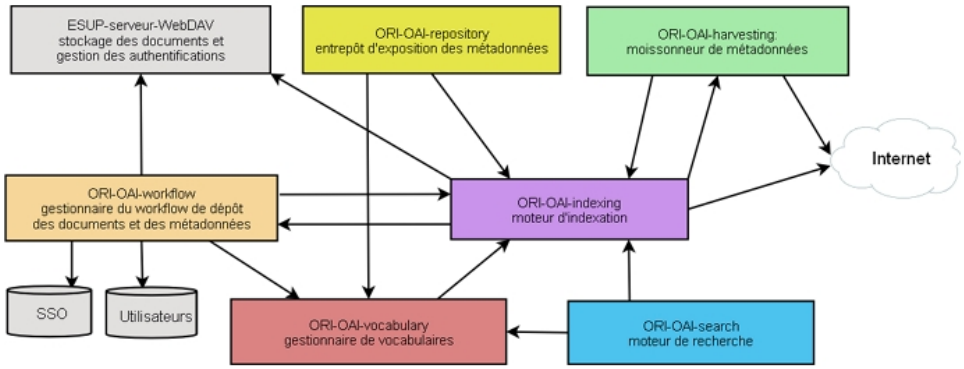
- Indexing dans Ori-Oai
- Lucene et applications attachées
- OSCache

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

Indexing dans ORI-OAI

ORI-OAI-indexing permet de gérer l'indexation et la recherche de documents locaux et distants. Il est appelé par le workflow via le frontal d'indexation pour indexer des fiches locales. En les indexant il reçoit un identifiant ainsi que d'autres éléments pour chacune d'elles. Le moissonneur fait également appel à lui, toujours par l'intermédiaire du frontal, en ce qui concerne l'indexation. ORI-OAI-indexing indexera une chaîne de caractères contenant la fiche. Le moissonneur envoie d'ailleurs un identifiant qui sera indexé avec la fiche pour faire le lien entre les deux modules.

Les recherches dans l'index sont effectuées par le module ORI-OAI-Search. Ce composant envoie une requête de type Lucene au module d'indexation. Il récupère ensuite de la part de ce dernier les attributs qui correspondent à sa recherche tels que le titre ou l'auteur par exemple.



This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

Lucene et applications attachées

Lucene est un moteur de recherche appartenant à la fondation apache permettant l'indexation et la recherche de texte. Il est entièrement écrit en langage Java. La version utilisée dans cette version du module d'indexation est la 2.3.2. Le lien vers le projet est le suivant : http://lucene.apache.org/java/docs/.

LIUS

LIUS, qui signifie Lucene Index Update and Search, est un framework d'indexation basé sur le projet Jakarta Lucene. Il a été développé à partir d'un ensemble de technologies JAVA et d'applications entièrement "open source". La documentation vers LIUS est donnée par le lien suivant : http://www.doculibre.com/lius/doc-1.0_fr.html

LIUS ajoute à Lucene plusieurs fonctionnalités d'indexation de type de documents tel que : Ms Word, Ms Excel, Ms PowerPoint, RTF, PDF, XML, HTML, TXT, la suite Open Office et les JavaBeans. Cet outil permet également d'effectuer une indexation mixte, qui a pour but d'intégrer tout le contenu d'un répertoire sous la même occurrence. Ceci est très utile lorsque l'utilisateur veut indexer des métadonnées en XML et le texte intégral en PDF ou dans un autre format. Ceci permet par la suite d'effectuer par exemple des recherches sur le titre, auteur et le texte intégral en même temps.

Toute la configuration de l'indexation, telle que le type de fichiers à indexer ou encore les champs par exemple, ainsi que la recherche sont définies dans un fichier XML, il ne reste plus qu'à écrire le code pour exécuter l'indexation ou la recherche.

Luke

Luke (http://www.getopt.org/luke/) est une interface graphique permettant de visualiser un index. Il peut être utile en tant qu'outil de diagnostic de ce dernier.

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

OScache

Il s'agit d'une librairie Open Source qui propose une solution de cache en utilisant des politiques de mise à jour (automatique ou manuelle) et de stockage (en mémoire ou sur disque) très flexibles. Il se présente sous trois formes : filtre de servlet, taglib JSP et API. Cette application a été créée par OpenSymphony et est consultable à l'adresse suivante : http://www.opensymphony.com/oscache/. OScache est utlisée dans le cadre du projet ORI-OAI-Indexing en ce qui concerne la gestion des différents caches utiles à l'optimisation de la recherche dans l'index.

Installation

This page last changed on Nov 25, 2008 by ycolmant@univ-valenciennes.fr.

- Introduction
- Configuration pour un deploiement rapide
- Premiers tests
- Personnalisation de la configuration

This page last changed on Oct 21, 2008 by ycaillau@univ-valenciennes.fr.

Introduction

Pré-Requis

- ORI-OAI-Indexing est une application fonctionnant avec le langage Java. Le JDK (version 5 ou ultérieure) doit donc être installé sur la machine de déploiement.
- Les tâches de compilation, de déploiement et certaines actions utilisent ANT.
- Une version de Tomcat doit être disponible sur la machine de déploiement. Le module a été testé avec la version 5.5.25 de Tomcat.

> Il est nécessaire de spécifier au Tomcat que vous utilisez l'encodage UTF-8 pour tous les modules. Pour cela, éditez le fichier PATH_TOMCAT_INDEXING/bin/startup.sh (startup.bat sous Windows) ou PATH_TOMCAT_INDEXING/bin/catalina.sh (catalina.bat sous Windows) et y ajoutez la commande suivante:
>
> - export CATALINA_OPTS="-Dfile.encoding=UTF-8 $CATALINA_OPTS" (sous Unix)
> - set CATALINA_OPTS="-Dfile.encoding=UTF-8 %CATALINA_OPTS%" (sous Windows)

Modifications par rapport à la version précédente

Le module d'indexation a subi de nombreuses modifications et ajouts.

> L'index créé à partir de la version 1.0 n'est pas compatible avec la version 1.1. Il faut donc le supprimer et utiliser la fonction de réindexation complète dans les modules Ori-Oai-Workflow et Ori-Oai-Harvester.

Nouvelles fonctionnalités

Dans cette version la fiche est directement stockée de manière compressée dans l'index. Le champ correspondant est md-ori-oai-notice-content.

Par ailleurs la suppression d'une fiche entraine l'optimisation de l'index mais la suppression de plusieurs fiches grâce à la méthode deleteNotices (voir la section "Méthodes publiques" de la partie utilisation pour plus de renseignements) entraine l'optimisation de l'index à la dernière fiche supprimée.

Une page d'accueil (voir le paragraphe "Lancement du module" de la section "Configuration pour un déploiement rapide") ainsi que des pages de visualisation de l'index ont été créées (voir la section "Administration" partie utilisation).

Un système de surlignage des résultats d'une recherche a été créé.

La commande ant init permet de supprimer de répertoire contenant l'index ainsi que le répertoire tmp et de les récréer.

Une recherche par attributs est capable de renvoyer tous les résultats dans le cas où les paramètres firstDoc et lastDoc sont à -1. Les performances de cette recherche dépendent donc du nombre de résultats.

Certains xpaths du fichier liusConfig.xml ont été modifiés dans le format LOM pour prendre en charge le LOM-fr.

Le module d'indexation ferme les IndexReader de Lucene laissés ouverts pour permettre une meilleure stabilité de l'application.

Le tri des éléments d'une recherche a été amélioré.

Fichier de configuration

Le ficher de configuration a été découpé en trois parties pour plus de clarté (voir la section "Personnalisation de la configuration").

Librairies

Afin de mettre en place le système de surlignage des résultats le jar de Lius a été modifié et renommé en Lius-ORI-1.0.jar.

Les librairies de Lucene dans sa version 2.3.2 ont remplacé les anciennes.

Le module d'indexation utilise aujourd'hui la librairie ori-oai-commons (voir la "Connexion au Web Service" de la partie utilisation); elle ne doit donc impérativement pas être supprimée du projet.

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

Configuration pour un déploiement rapide

Il existe plusieurs modes d'installation du module ori-oai-indexing. Tout d'abord en utilisant ori-oai-commons-quick-install. Ceci vous permettra de déployer la suite ori-oai avec un minimum de personnalisation tout ceci en utilisant un seul fichier de configuration. L'installation manuelle vous fera éditer manuellement différents fichiers afin de configurer au mieux votre application. Il est préférable d'utiliser la première solution. En effet, celle-ci vous apportera un déploiement rapide de ORI-OAI sur un serveur de production avec une configuration de base. Vous pourrez toutefois après cette installation apporter toutes les configurations avancées que vous souhaitez à vos modules.

Installation rapide avec ori-oai-commons-quick-install

Ce mode d'installation permet une installation rapide de ORI-OAI en saisissant tous les paramètres généraux aux modules dans un seul fichier de configuration. Vous pourrez compléter votre installation et vos paramétrages par la suite sans aucun souci. Reportez-vous à la documentation en ligne à l'adresse: http://sourcesup.cru.fr/ori-oai-commons/quick-install

Installation manuelle

init-build.properties

Ce fichier se trouve à la racine du repertoire ori-oai-indexing. Il contient les données nécessaires au déploiement correct de l'application dans Tomcat. Copiez ce fichier en build.properties. Ouvrez le nouveau fichier créé et modifiez-le comme indiqué ci-dessous.

> Ne supprimez pas le fichier init-build.properties. Il est utilisé dans le cadre d'une installation rapide avec ori-oai-commons-quick-install.

Le fichier se présente de la manière suivante :

```
############################
##                        ##
##   PARTIE INSTALLATION  ##
## 2 modes d'installation ##
##        1) ou 2)        ##
##                        ##
############################


#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
# 1) Utilisation de ori-oai-commons-quick-install
#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

#URL du fichier contenant toutes les proprietes pour ce module en installation rapide
#Commentez le parametre si vous ne voulez pas utiliser les fonctionnalites d'installation de ori-oai-commons-quick-install
commons.parameters.central.file.url=[COMMONS_PARAMETERS_CENTRAL_FILE_URL]


#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
## 2) Installation manuelle du module
# Dans ce cas, il est necessaire de commenter
# le parametre commons.parameters.central.file.url
#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#
```

```
# Cette partie du fichier doit etre mise a jour avant
# la premiere utilisation de votre application dans votre environnement

#Repertoire d'installation de Tomcat
tomcat.home  = [PATH_TOMCAT_INDEXING]

#Repertoire de deploiement
deploy.home  = [PATH_TOMCAT_INDEXING]/webapps/

#Nom de ditribution de l'application
app.name.deploy=[CONTEXT_INDEXING]

#URL vers Tomcat
tomcat.URL.deploy = http://[HOST_INDEXING]:[PORT_INDEXING]

#Dossier dans lequel se trouve l'index
index.directory = [INDEXES_DATA_DIR]/index-indexing/index

#Dossier dans lequel se trouvent les fichiers temporaires
temp.directory = [INDEXES_DATA_DIR]/index-indexing/tmp
```

Etant donné que vous faites une installation manuelle, il est nécessaire de commenter la variable commons.parameters.central.file.url dans ce fichier comme ceci:

```
#URL du fichier contenant toutes les propriétés pour ce module en installation rapide
#Commentez le parametre si vous ne voulez pas utiliser les fonctionnalites d'installation de ori-oai-commons-quick-install
#commons.parameters.central.file.url=[COMMONS_PARAMETERS_CENTRAL_FILE_URL]
```

tomcat.home

> Répertoire racine de Tomcat. Remplacez PATH_TOMCAT_INDEXING par le chemin où se trouve le serveur Tomcat où sera installé le module ori-oai-indexing

deploy.home

> Répertoire dans lequel sera déployée l'application (il doit s'agir du répertoire *webapps* de Tomcat). Remplacez PATH_TOMCAT_INDEXING par la même valeur que celle donnée dans tomcat.home

app.name.deploy

> Nom de l'application dans le contexte Tomcat. Remplacez CONTEXT_INDEXING par le nom que vous souhaitez donner au répertoire de déploiement du module (exemple : ori-oai-indexing ou indexing). Ce nom servira également à retrouver la page d'accueil du module qui sera : "HOST_INDEXING:PORT_INDEXING/CONTEXT_INDEXING/"
> (voir section 4 de la page utilisation).

tomcat.URL.deploy

> URL vers la page d'accueil de Tomcat. Ce champ est utile dans le cadre des Premiers Tests. Remplacez HOST_INDEXING et PORT_INDEXING par le nom de la machine et le numéro de port du Tomcat où est installé le module ori-oai-indexing.

index.directory

> Répertoire dans lequel se trouve l'index.

temp.directory

> Répertoire dans lequel se trouvent les fichiers temporaires.

log4j.properties

Le fichier log4j.properties se présente de la manière suivante :

```
# ----- PARTIE A MODIFIER PAR L'UTILISATEUR
log4j.appender.fichier.file=[PATH_TOMCAT_INDEXING]/logs/ori-oai-indexing.log

# ----- Definition des logs fichier
log4j.logger.org.orioai.indexing=INFO,fichier
#log4j.appender.fichier=org.apache.log4j.ConsoleAppender
log4j.appender.fichier=org.apache.log4j.FileAppender

log4j.appender.fichier.layout=org.apache.log4j.PatternLayout
log4j.appender.fichier.layout.ConversionPattern=%5p %d{MMM/dd HH:mm:ss} %c :: %m%n
```

Les paramètres à modifier éventuellement sont les suivants :

log4j.logger.org.orioai.indexing

Niveau de logs Trois niveaux sont utilisés dans ORI-OAI-Indexing.

- ERROR : C'est le niveau à utiliser en production. Seules les erreurs seront notifiées dans le fichier pour optimiser au maximum les performances.
- INFO : Il contient les erreurs mais aussi les informations sur l'identifiant des fiches indexées ainsi que les requetes lancées et le nombre de résultats correspondants.
- DEBUG : Version de deboggage en cas de problème avec le module. A ce niveau les performances de l'indexeur sont plus basses.

log4j.appender.fichier.file

Emplacement et nom du fichier de logs. Remplacez PATH_TOMCAT_INDEXING par le chemin où se trouve le serveur Tomcat où sera installé le module ori-oai-indexing.

configIndexing.xml

Il reste une dernière étape avant de déployer l'application. Il s'agit ici de remplir le fichier configIndexing.xml situé dans le dossier "properties". Pour cela il suffit simplement de remplacer INDEXES_DATA_DIR dans les balises "indexDir" et "tmpDir". Il faut également modifier PROXY_HOST et PROXY_PORT par la valeur adéquate. Si vous n'utilisez pas de proxy il suffit alors de supprimer ces valeurs. La section 5.1 offre plus de renseignements sur ce fichier ce qui permet une configuration plus poussée du module.

Déploiement de l'application

- Si c'est votre premier déploiement ou si vous souhaitez supprimer un index existant lancez : **ant init**

Un message vous previendra que vous tentez de supprimer l'index. Appuyez sur "y" puis la touche "Entrée" pour continuer l'initialisation des répertoires.

- Lancez la commande **ant all** pour compiler les fichiers sources et créer le contexte du module d'indexation dans le serveur Tomcat.

Lancement du module

Il ne reste plus qu'à demarrer Tomcat pour lancer ORI-OAI-Indexing. Vous pouvez vérifier si le module fonctionne en testant sur un navigateur l'URL suivante : HOST_INDEXING:PORT_INDEXING/ori-oai-indexing/" en modifiant "HOST_INDEXING:PORT_INDEXING"

par la valeur adéquate. Vous devriez obtenir un affichage similaire à celui-ci :


Le module est complètement chargé au bout de quelques secondes. S'il est sollicité trop tôt, une erreur 404 survient alors.


Les autres onglets ne fonctionnent que si l'index est créé. Ils ne sont donc pas utilisables à cet instant.

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

Premiers tests

Si le module est correctement installé lancez le tomcat :

```
[ORI_HOME]/tomcat-indexing/bin/startup.sh
```

Ouvrez un navigateur et tapez l'url :

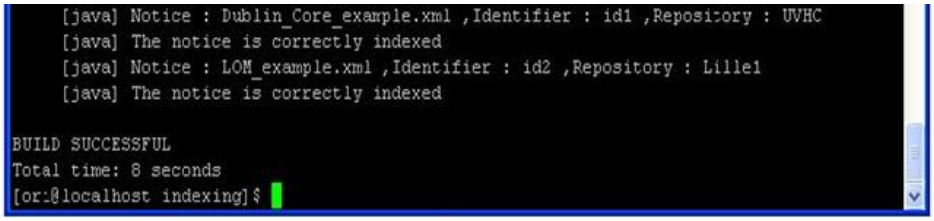
```
http://[HOST_INSTALL]:8182/ori-oai-indexing/
```

Vous devriez obtenir l'affichage suivant :

Pour vérifier si le module fonctionne correctement, placez-vous dans le répertoire **[ORI_HOME]/src/ori-oai-indexing-svn** puis tapez :

```
ant testIndex
```

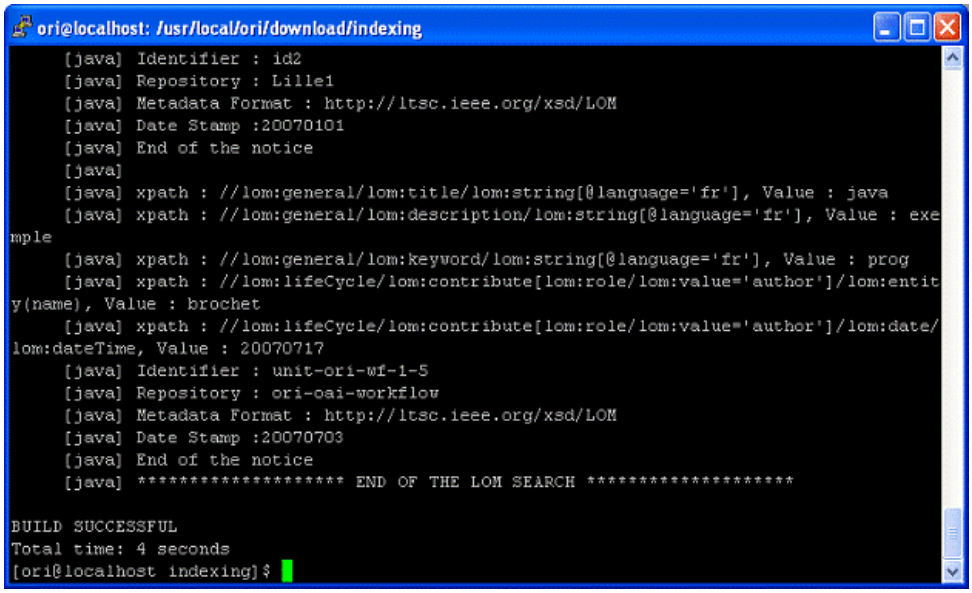
Deux documents vont alors être indexés. Si l'indexation s'est bien passée vous devriez avoir :



Il reste à vérifier si la recherche fonctionne correctement. Pour cela tapez :

ant testSearch

Vous devriez alors obtenir le résultat suivant :



Si tout s'est correctement déroulé, il vous faut supprimer cet index de test en supprimant le contenu du dossier nommé **index** dans **[ORI_HOME]/data/index-indexing** afin que les fiches de test n'apparaissent pas dans votre index de production. Pour cela, arrêtez votre tomcat, supprimez le contenu de **index** et redémarrez le tomcat. Pour cela, vous pouvez lancer la tâche **ant init** depuis **[ORI_HOME]/src/ori-oai-indexing-svn**

NB : Vous pouvez visualiser votre index en vous rendant à la page **http:// [HOST_INSTALL] :8182/ori-oai-indexing/** et en consultant l'onglet "Visualisation de toutes les pages". Si vous cliquez sur l'identifiant d'une fiche, vous verrez alors apparaitre toutes les métadonnées indexées de celle-ci.

Important : Il est fortement recommandé de sauvegarder régulièrement votre index en copiant le dossier **index**. Si votre index devenait inutilisable, il vous suffirait alors de supprimer le dossier index et de le remplacer par votre copie la plus récente. Un redémarrage de votre serveur Tomcat hébergeant le module d'indexation sera nécessaire.

Il existe également des procédures de restauration de l'index depuis les modules ori-oai-workflow et ori-oai-harvester.

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

Personnalisation de la configuration

Configuration générale

Le fichier configIndexing.xml contient la configuration générale de l'application. Il se trouve dans le dossier *properties*. Il se décompose en trois sections :

- Partie à modifier par l'utilisateur. C'est dans cette section que l'utilisateur indiquera notamment le répertoire où sera créé l'index ainsi que les paramètres du proxy.
- Options modifiables par l'utilisateur. Cette section propose divers éléments que l'utilisateur peut modifier s'il le souhaite comme la durée des caches ou encore le nombre de documents à indexer avant d'optimiser l'index.
- Partie à ne pas modifier par l'utilisateur. Cette section ne doit en aucun cas être modifiée car cela engendrerait des dysfonctionnements graves du module.
  Le fichier se présente de la manière suivante :

```
<config>

    <!-- PARTIE A MODIFIER PAR L'UTILISATEUR -->

        <!-- Repertoire de l'index -->
        <indexDir>[INDEXES_DATA_DIR]/index-indexing/index</indexDir>

        <!-- Repertoire des fichiers temporaires -->
        <tmpDir>[INDEXES_DATA_DIR]/index-indexing/tmp</tmpDir>

        <!-- Proxy -->
        <proxy>
            <proxyHost>[PROXY_HOST]</proxyHost>
            <proxyPort>[PROXY_PORT]</proxyPort>
        </proxy>

        <!-- Fichier de configuration LIUS -->
        <liusConfigFile>liusConfig.xml</liusConfigFile>


    <!-- OPTIONS MODIFIABLES PAR L'UTILISATEUR -->

        <!-- Frequence d'optimisation de l'index -->
        <frequencyOfOptimization>200</frequencyOfOptimization>

        <!-- Classe de transformation de requete -->

 <queryTransformerClass>org.orioai.indexing.search.querytransformer.AccentRemoverTransformation</
queryTransformerClass>

        <!-- Duree de vie des elements de chaque cache -->

            <!-- Cache 1 en secondes comportant les TreeSet des ValeurRetrouvee -->
            <attributesCache>900</attributesCache>

            <!-- Cache 2 en secondes pour la gestion des fichiers temporaires -->
            <noticesCache>900</noticesCache>

            <!-- Cache 3 en secondes du nombre de resultats d une requete -->
            <nbResultsCache>1800</nbResultsCache>
```

```
        <!-- Formats de métadonnées -->
        <format id="dublin_core">
            <namespace prefix="dc" uri="http://purl.org/dc/elements/1.1/"/>
            <namespace prefix="oaidc" uri="http://www.openarchives.org/OAI/2.0/oai_dc/"/>
            <xpathToPlainText>//dc:relation</xpathToPlainText>
        </format>

        <format id="formation">
            <namespace prefix="cdm" uri="http://www.w3.org/2001/XMLSchema-instance"/>
            <xpathToPlainText />
        </format>

        <format id="pedagogique">
            <namespace prefix="lom" uri="http://ltsc.ieee.org/xsd/LOM"/>
            <xpathToPlainText>//lom:technical/lom:location</xpathToPlainText>
        </format>

        <!-- Chaines de remplacement -->
        <replacement stringToReplace=":" stringReplacement="@">
            <metadata>md-ori-oai-id</metadata>
            <metadata>md-ori-oai-namespace</metadata>
        </replacement>

        <replacement stringToReplace=" " stringReplacement="_">
            <formatId>pedagogique</formatId>
            <xpath>//lom:classification/lom:taxonPath[lom:source/lom:string='dewey']/lom:taxon/
lom:id</xpath>
        </replacement>

        <replacement stringToReplace="vCardToUpperCaseORI" stringReplacement="vCardToUpperCaseORI">
            <formatId>pedagogique</formatId>
            <xpath>//lom:contribute/lom:entity</xpath>
        </replacement>


    <!-- PARTIE A NE PAS MODIFIER PAR L'UTILISATEUR -->

        <!-- Nom des metadonnees communes aux modules ORI-OAI -->
        <static_metadatas>
            <doc_id>md-ori-oai-id</doc_id>
            <repository>md-ori-oai-repository</repository>
            <format>md-ori-oai-namespace</format>
            <datestamp>md-ori-oai-datestamp</datestamp>
            <score>md-ori-oai-score</score>
            <workflow_name>ori-oai-workflow</workflow_name>
            <notice_content>md-ori-oai-notice-content</notice_content>
            <crawled>md-ori-oai-crawled</crawled>
        </static_metadatas>

        <!-- Tests du crawler web (en cours de developpement) -->

            <!-- Planification du crawler -->
            <scheduleCrawler />

            <!-- Valeurs possibles du champ crawled -->
            <staticCrawledValues>
                <notCrawled>no</notCrawled>
                <notSuccessful>failed</notSuccessful>
                <forbidden>forbidden</forbidden>
                <successful>yes</successful>
            </staticCrawledValues>
```

```
</config>
```


Les parties static_metadatas ainsi que staticCrawledValues ne doivent pas être modifiées.

Le fichier se configure comme ceci:

indexDir

Chemin vers le répertoire dans lequel se trouve l'index, ici il s'agit de "ORI_HOME/data/indexes/index". Le répertoire "ORI_HOME/data/indexes" doit impérativement être créé mais le répertoire "Index" pourra normelement être créé par l'application au moment de l'indexation mais Il est possible en cas de problèmes que ce répertoire soit créé à la main.

tmpDir

Répertoire dans lequel se trouveront les fichiers temporaires, qui servent en réalité du cache, utiles à la récupération de fiches. Le dossier doit être déjà créé. Les fiches n'y seront pas directement stockées mais elles seront dans un sous-dossier appelé "ori-oai-indexing".

proxyHost

Paramètre du proxy. Si l'application ne passe pas par un proxy, il suffit d'effacer le contenu de la balise.

proxyPort

Port du proxy. Si l'application ne passe pas par un proxy, il suffit d'effacer le contenu de la balise.

liusConfigFile

Fichier de configuration LIUS qui sera utilisé par ORI-OAI-Indexing. Par défaut il s'agit de liusConfig.xml qui fonctionne à la base avec les formats DC, LOM et CDM. Ici on ne renseigne pas le chemin mais simplement le nom du fichier. C'est pourquoi le fichier doit obligatoirement se trouver dans le répertoire properties. Pour plus d'informations sur ce fichier veuillez consulter la section 5.2.

frequencyOfOptimization

Permet d'optimiser l'index dès que l'on a atteint le nombre de fiches indiqué. Il est préférable de ne pas dépasser 500 sous peine de rendre l'index inutilisable.

queryTransformerClass

Chemin vers la classe permettant de modifier la requete pour retrouver plus de résultats pertinents dans l'index. La classe donnée par défaut sert à supprimer les accents dans la requête pour gérer au mieux les caractères spéciaux de la langue française. Il est possible de créer une nouvelle classe qui héritera de la classe abstraite "QueryTransform" dans le but de modifier autrement la requête.

attributesCache

Durée de vie en secondes du cache conservant les résultats de la recherche par attributs.

noticesCache

Durée de vie en secondes du cache des fichiers temporaires

nbResultsCache

Durée de vie en secondes du cache conservant le nombre de résultats d'une requête donnée.

format

Contient le namespace et le prefix d'une format de métadonnée.

replacement

Contient les caractères à modifier dans une chaine provenant d'une métadonnée supplémentaire (md-ori-oai-...) ou provenant d'un xpath d'un format précis.

Le couple "format - replacement" permet de gérer au mieux l'indexation et la recherche de documents. Pour comprendre au mieux son utilité prenons l'exemple du code Dewey. Il s'agit en réalité d'une donnée composée de chiffres et d'espaces (ex : 125 17.1). Or cette donnée n'est pas indexée correctement à cause du caractère d'espacement. Donc pour résoudre ce problème, ce caractère est remplacé par un autre caractère, ici "_" au moment de l'indexation. Pour remplacer un caractère par un autre, deux éléments sont nécessaires; le premier est le ou les formats concernés par le remplacement. Le second est le xpath permettant de retrouver la donnée à modifier. Dans notre exemple le format est le LOM dont l'identifant de format est "pédagogique". La donnée indexée sera alors "125_17.1". Il faut noter que ces remplacements de caractères ne modifient en rien la forme de la requête lors d'une recherche, c'est-à-dire dans notre cas que l'on recherche le code Dewey "125 17.1" et non pas "125_17.1".

L'identifiant et le namespace sont modifiés lors de l'indexation. En effet le caractère ":" est remplacé par "@". Ceci est utile dans le cadre d'une recherche. Un requête de type Lucene utilise le caractère ":" pour séparer l'attribut de la valeur (ex: "nom : toto"). Donc la valeur ne peut pas comporter la caractère ":" ce qui explique la modification. Mais du point de vue de l'utilisateur, il n'y a aucune modification de caractères à réaliser. Ceci ne sert qu'en interne au module d'indexation et est donc transparent pour tous les autres modules.

Configuration de Lius

Le fichier de configuration de LIUS, appelé liusConfig.xml, se trouve dans le dossier "Properties".

Selection de l'analyseur

Il permet d'indexer le contenu dans une langue donnée. Il est défini dans la balise "analyzer", elle-même incluse dans la balise "properties". Par défaut, c'est l'analyseur de la langue française qui est choisi. Cet analyseur a été modifié dans le cadre du projet ORI-OAI pour prendre en compte plus de spécificités manquantes de la langue françaises comme les mots composés par exemple.

Les autres éléments de la balise "properties" n'ont pas à être modifés.

Ajout d'un nouveau format xml

Ceci peut-être utile si vous souhaitez indexer un format différent du DC, du LOM ou du CDM, tous trois déjà présents par défaut. Pour se faire, ajoutez dans la balise "xml" le code suivant :

```
<xmlFile ns="[namespace]" setBoost="[valeur]">
    <indexer class="org.orioai.indexing.index.indexer.OriOaiXmlFileIndexer">
        <mime>text/xml</mime>
    </indexer>
    <fields>
    </fields>
</xmlFile>
```

namespace

Il s'agit du namespace correspondant au format

valeur

Valeur décimale de boost. Le maximum est 2.0. L'attribut setBoost permet de prioriser un format par rapport à un autre. Cet attribut est optionnel; si il n'est pas ajouté, la valeur de

boost sera 1.0. Le boost a une influence sur le calcul de la pertinence d'un résultat de la recherche


Ajouter un nouveau format ne suffit pas à indexer le contenu d'une fiche. Il faut également ajouter les xpaths à indexer pour le format (voir section 5.2.3).

Ajout d'un nouveau xpath à indexer dans le format

Ceci peut-être utile si vous souhaitez indexer une métadonnée différente de celles présentes par défaut dans les formats DC, LOM, CDM ou d'un nouveau format que vous avez créé. Pour se faire, trois étapes sont nécessaires :

- Dans la balise "fields" du format ajoutez ceci :

```
<luceneField name="[xpath_encodé_UTF8]" xpathSelect="[xpath]" type="[type]"
 setBoost="[valeur]"/>
```

xpath

Il s'agit du xpath vers la métadonnée que l'on souhaite indexer.

xpath_encodé_UTF8

Le nom de ce champ est par convention dans ORI-OAI le xpath qui a été encodé grâce à la méthode java.net.UrlEncoder.encode.

type

Cet attribut indique le type de métadonnée. Plusieurs types sont possibles : "Text", "concatDate" s'il s'agit d'une date et "vcard" si la métadonnée contient une vCard.

valeur

Valeur décimale de boost. Le maximum est 2.0. L'attribut setBoost permet de prioriser une métadonnée par rapport à une autre. Cet attribut est optionnel; s'il n'est pas ajouté, la valeur de boost sera 1.0. Le boost a une influence sur le calcul de la pertinence d'un résultat de la recherche


L'attribut "setBoost" peut être à la fois dans la balise "format" et dans la balise "luceneField" d'un format.

- Dans la balise "searchFields" elle-même incluse dans la balise "multiFieldQueryParser", créez une nouvelle ligne et ajoutez-y xpath_encodé_UTF8 précédé d'une virgule.


La virgule est indispensable au bon fonctionnement du module.

- Dans la balise "fieldsToDisplay" elle-même incluse dans la balise "searchResult", ajoutez ceci :

```
<luceneField name="[xpath_encodé_UTF8]" label="[xpath]"/>
```

xpath

Il s'agit du xpath vers la métadonnée que l'on souhaite indexer.

xpath_encodé_UTF8

Le nom de ce champ est par convention dans ORI-OAI le xpath qui a été encodé grâce à la méthode java.net.UrlEncoder.encode.

Système de surlignage

Il permet de mettre en valeur les résultats de votre recherche.
Pour activer/désactiver ce système il suffit de remplir l'attribut setHighlighter de la balise fieldsToDisplay à true/false (vers la fin du fichier).

Utilisation

This page last changed on Nov 25, 2008 by ycolmant@univ-valenciennes.fr.

- Connexion au Web Service
- Methodes publiques
- La classe SearchResults
- Administration
- Contraintes d'utilisation

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

Connexion au Web Service

Voici la méthode permettant de se connecter au module d'indexation :

```
private static OriOaiIndexingServiceInterface getService(String url) throws Exception {
    ObjectServiceFactory objectServiceFactory = new ObjectServiceFactory();
    Class classe = OriOaiIndexingServiceInterface.class;
    Service serviceModel = objectServiceFactory.create(classe);
    OriOaiIndexingServiceInterface port = (OriOaiIndexingServiceInterface) new XFireProxyFactory().create(serviceModel, url);

    return port;
}
```

Il ne reste plus qu'à ajouter dans le code l'appel à cette méthode en donnant comme URL celle du Web Service de l'indexeur : OriOaiIndexingServiceInterface service = getService(url);

Ori-Oai-Indexing intègre la librairie ori-oai-commons.jar. Elle est indispensable pour se connecter au Web Service.

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

Méthodes publiques du moteur d'indexation

IndexOrUpdate

Indexation d'une fiche s'il s'agit de la première indexation de celle-ci ou mise à jour si elle est déjà présente. Elle est utilise pour ORI-OAI-Workflow car ce module ne gère pas la présence ou non d'une fiche dans l'index. L'index sera toujours optimisé lors de l'utilisation de cette méthode.

Les paramètres de cette méthode sont les suivants :

String metadataFile

> Fiche de métadonnées à indexer

String id

> Identifiant associé à la fiche

String namespace

> Namespace de la fiche

String datestamp

> Date au format YYYY MM DD. Le caractère de séparation sera supprimé au moment de l'indexation et la date sera de la forme : YYYYMMDD. Il n'est pas possible d'indexer une objet de type date.

String repository

> Repository dans lequel se trouve la fiche. Si c'est une fiche locale, la valeur de ce paramètre sera "null".

Cette méthode renvoie un entier indiquant si la procédure s'est effectuée correctement (1) ou si un problème est survenu (-1).

Index

Indexation classique d'une fiche. Elle est utilisée par ORI-OAI-Harvester pour indexer plusieurs fiches car les performances de cette méthode sont plus intéressantes que celles d'indexOrUpdate.

Un paramètre est ajouté par rapport à indexOrUpdate :

Boolean doOptimize

> Booléen à true lorsque l'index doit être optimisé. L'optimisation est une phase relativement longue (selon la taille de l'index). Lorsque plusieurs fiches sont indexées, doOptimize ne sera à true que lorsque l'on indexera la dernière fiche.

Cette méthode renvoie un entier indiquant si l'indexation s'est effectuée correctement.

DeleteNotice

Suppression d'une fiche dans l'index.

Les paramètres sont :

String id

Identifiant de la fiche à supprimer.

String namespace

Namespace de la fiche. Un même identifiant peut-être utilisé pour plusieurs fiches de format différent. Si namespace est "null", on supprime toutes les fiches comportant l'identifiant donné.

Cette méthode renvoie un entier indiquant le bon fonctionnement de la suppression.

DeleteNotices

Suppression de plusieurs fiches de l'index. Pour chaque fiche à suprimer elle appelle la méthode deleteNotice Elle prend en paramètres :

String []ids

Tableau de d'identifiants à supprimer

String []namespaces

Namespaces associés aux identifiants.

Cette méthode renvoie un tableau d'entiers donnant le résultat de chaque suppression.

Update

Mise à jour d'une fiche dans l'index. Cette méthode est utilisée par ORI-OAI-Harvester car il gère la présence d'une fiche dans l'index. La fiche est d'abord supprimée de l'index puis réindéxée avec les nouveaux paramètres. Ces derniers sont identiques à ceux de la méthode index.

Cette méthode renvoie un entier indiquant le bon fonctionnement de la suppression.

SearchForNumberOfResults

Nombre de résultats pour une requête donnée.

String request

Requete dont on cherche a connaitre le nombre de résultats

Cette méthode renvoie un entier long correspondant au nombre de résultats d'une requête.

SearchForSomeNumberOfResults

Nombre de résultats pour un tablau composé de plusieurs requêtes.

String request[]

Tableau de requêtes dont on cherche a connaitre le nombre de résultats

Cette méthode renvoie un tableau d'entiers long correspondant au nombre de résultats de chaque requête.

SearchXMLDoc

Récupère une fiche XML locale ou moissonnée.

Elle prend comme paramètres :

String id

Identifiant de la fiche.

String repository

Repository de la fiche. Ce paramètre indique qu'il s'agit d'une fiche locale ou moissonnée. Si la fiche est locale, ce paramètre vaut "workflow".

String namespace

Namespace de la fiche à récupérer.

Cette méthode renvoie une chaîne de caractères correspondant à la fiche XML.

SearchXMLDocs

Récupère plusieurs fiches XML.

Les paramèrtes sont :

String request

Requête dont on souhaite obtenir les fiches correspondant aux résultats.

int firstDocumentId

Numéro du premier document dans la liste des résultats à renvoyer.

int lastDocumentId

Numéro du dernier document dans la liste des résultats à renvoyer.

Cette méthode renvoie un objet de type SearchResults. Cette classe est présentée dans la section suivante.

SearchFromAttributes

Recherche par attributs. Ici on ne récupère pas de fiches mais certains attributs (ex : titre, auteur..) de cette dernière.

Les paramètres de cette méthode sont :

String request

Requête dont on souhaite connaitre les résultats.

String sortAttributes[]

Attributs de tri de la liste de résultats. Si cette valleur est "null" alors les résultats seront triés par leur identifiant.

int firstDocumentId

Numéro du premier document dont on souhaite connaitre les valeurs des attributs.

int lastDocumentId

Numéro du dernier document dont on souhaite connaitre les valeurs des attributs.

String []attributes

Attributs dont on souhaite connaitre la valeur.

boolean ascending

Booléen à true si les résultats doivent être rangés dans l'ordre croissant.

boolean highlightTerms

Booleen qui indique si on doit tenter de surligner la valeur des attributs.

Cette méthode renvoie un objet de type SearchResults. Cette classe est présentée dans la section suivante.

En donnant "-1" comme valeur à firstDocumentId et lastDocumentId, tous les résultats sont alors retournés. Dans ce cas le temps de réponse peut être plus long si le nombre de résultats est important.

uniqueValues

Recherche de valeurs uniques de l'index. Elle est utilisée pour le recherche par auteur notamment.

String xpath

Xpath ou nom de métadonnée dont on souhaite connaitre toutes les valeurs uniques.

Cette méthode renvoie un tableau de chaînes de caractères correspondant à toutes les valeurs uniques.

clearCache

Méthode qui vide les cache sans avoir à redémarrer Tomcat. Elle est utilisée en cas de tests de recherche.

String name

Nom du cache à vider. S'il vaut null alors tous les caches sont vidés.

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

La classe SearchResults

Elle est utilisée lors de la recherche par attributs ou par la recherche de fiches XML. Elle contient un *entier
long *correspondant au nombre de résultats d'une requête ainsi qu'une *liste d'objets de type SearchResult*. Pour les obtenir il faut respectivement utiliser les méthodes **getNumberOfResults()** et **getResults()**.

La classe SearchResult contient plusieurs chaînes de caractères : id, namespace, repository, datestamp que l'on récupère grâce à des méthodes get. Elle contient également une chaîne correspondant à la fiche et que l'on obtient grâce à la méthode **getNoticeContent()**. Elle est utilisée dans le cadre de la recherche de fiches XML. En ce qui concerne la recherche par attributs il faut utiliser *getAttributesValues()*qui renvoie une liste de liste de String. En effet chaque attribut peut avoir plusieurs valeurs (ex : plusieurs auteurs), et plusieurs attributs peuvent être demandés (ex : titre, auteur, description...). Donc l'élément renvoyé est bien une liste de liste de chaînes de caractères.

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

Administration du module d'indexation

Quelques fonctionnalités permettent de consulter l'index depuis le navigateur grâce à l'adresse suivante : http://HOST_INDEXING:PORT_INDEXING/CONTEXT_INDEXING.

- Accueil : Page d'accueil du module
- Visualisation de toutes les fiches : Montre toutes les fiches présentes dans l'index. En cliquant sur le l'identifiant d'une fiche, on peut en visualiser toutes ses données. On passe alors dans l'onglet "Visualistaion d'une fiche". Si on clique sur l'onglet "Visualisation d'une fiche" un message indique qu'il faut retrourner dans la page "Visualisation de toutes les fiches" et cliquer sur un identifiant pour voir la fiche complète.
- Recherche : Permet de lancer une requête Lucene

This page last changed on Oct 17, 2008 by ycaillau@univ-valenciennes.fr.

Contraintes d'utilisation

Pour un fonctionnement optimal du moteur d'indexation, certaines règles doivent être respectées.

- Lors d'une indexation, le paramètre datestamp doit être une chaîne de caractères de la forme YYYY/MM/DD. Le caractère de séparation n'a pas d'importance car il sera supprimé avant d'être indexé. Lors d'une recherche, la date aura alors la forme : YYYYMMDD.
- Pour indexer des fiches contenant des Vcards, il est indispensable que la chaîne de caractères correspondant à la fiche conserve l'indention de celle-ci. La chaîne de caractères ne doit donc pas être sur une seule ligne. Cette règle doit être également respectée lors de la récupération d'une fiche.
- Une requête doit avoir une forme bien précise : "xpath:(valeur)". L'utilisation des parenthèses est indispensable pour la bonne utilisation des requêtes. Voici un exemple de requete correcte : "md-ori-oai-id:(mon_identifiant OR mon_autre_identifiant) AND md-ori-oai-namespace:(mon_namespace)". De plus il ne doit pas y avoir d'espaces entre le xpath et le caractère de séparation ":".
- Le xpath d'une requête doit être encodé (grâce à la méthode java.net.URLEncoder.encode). Par exemple : "//dc:title:(mon_titre)" doit s'écrire : "%2F%2Fdc%3Atitle:(mon_titre)".

Il est impératif de sauvegarder très régulièrement l'index. Pour cela il faut conserver tout le répertoire contenant l'index. Il se trouve au niveau du chemin renseigné dans la balise *indexDir* du fichier configIndexing.xml.