Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg
University of Applied Sciences of Fribourg

Travail pratique
eXist-db

Houda CHABBI DRISSI

(Source d'inspiration de ce TP, sauf la partie Lucene :)
http://www.lif.univ-mrs.fr/~preynier/XML/tp7/tp7.html

1 Introduction

L'outil eXist-db est un des SGBD natif XML, open source, des plus complets, avec une communauté très active, facile à mettre en œuvre et qui permet à travers ces APIs de prototyper des sites web alternativement au site classique (par exemple MySQL + PHP) : (http://exist-db.org/exist/apps/doc/devguide.xml).

Nous allons voir comment implémenter une application web dédiée à des données XML et qui n'utilise que des technologies du monde XML à savoir : *eXist*, *XQuery* et un peu de XSLT et pour permettre des affichages à notre convenance. eXist-db offre 3 manières différentes d'envisager de monter un site web. Nous utiliserons dans ce TP celle basée sur le ***XQueryServlet.*** L'application que nous allons développer dans ce TP permet à un utilisateur d'administrer ces données XML à travers eXist-db :

1. Charger des documents XML,
2. Lancer des requêtes sur ses données (Xquery),
3. Modifier ses données (XQuery Update),
4. Appliquer des feuilles de style aux documents produits (XSLT).
5. Interroger la base de données via les index Lucene (~ Xquery FT)

2 Travail à réaliser

2.1 Installation de eXist-db

La première chose à faire est d'installer eXist. Pour cela, veuillez-vous référer au document en annexe.

2.2 Installation du site

Dans les exercices suivants, nous utiliserons comme exemple un site très simple permettant la gestion d'une base de données de films. Comme indiqué dans l'introduction, nous allons donc utiliser la technologie *XQueryServlet* de *eXist-db* pour réaliser ce site.

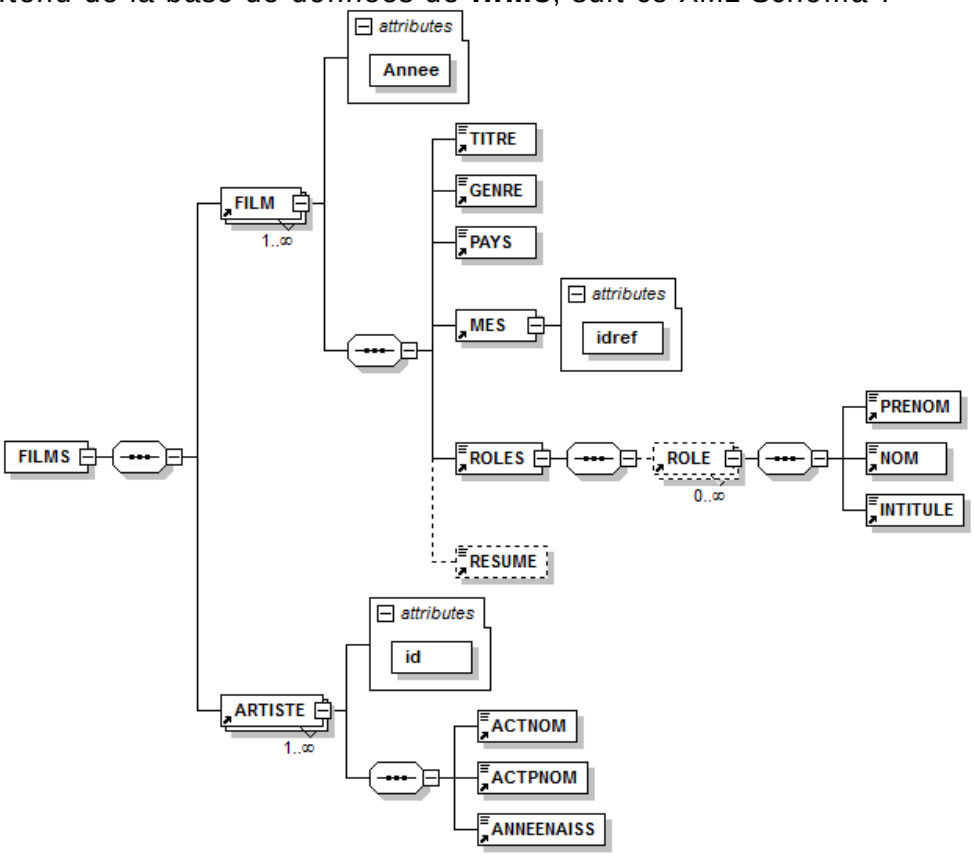
> ***XQueryServlet :*** *The XQueryServlet reads an XQuery script from the file system, usually from the directory in which the web application resides, and executes it with the current HTTP context. The result of the query is returned as the content of the HTTP response.*

Using the XQueryServlet is thus similar to calling a JSP page or PHP script, which resides on the file system of the server.
*The **XQueryServlet** is a servlet which responds to URL-patterns (e.g. *.xql and *.xqy) as defined in the web.xml configuration file of the application. The servlet will interpret this pattern as pointing to a valid XQuery file. The XQuery file is then loaded, compiled and executed with the current HTTP context. The results are then sent to the client in the content of the HTTP response.*

Les fichiers de ce site se trouvent dans les dossiers suivant :

- **« site.zip »** contenant les fichiers html du « Site ». A décompresser la où vous le souhaitez.
- **« BD.zip »** contient des fichiers XQuery pour la manipulation de la base de données**. Ce dossier décompressé doit être copié dans « <install_eXist> / webapp »** afin de permettre au ***XQueryServlet*** de les exécuter**.**

Les différents fichiers contenus dans ces deux dossiers seront modifiés tout au long des différents exercices de ce TP.
Le contenu de la base de données de **films**, suit ce XML Schema :



A noter que l'élément « MES » contenu dans films représente le réalisateur du film, identifié par l'attribut « IDREF » qui correspond à l'attribut « ID » d'un élément « Artiste »

Le contenu du fichier « BD/films.xml » est un exemple d'instance XML de ce Schema.

Compréhension du site mis à disposition

Le but de cet exercice est de tester le site de gestion de la BD de films et de comprendre sa structure.

Pour cela, ouvrer la page « site/index.html » dans votre navigateur. Cette page contient cinq fonctionnalités :

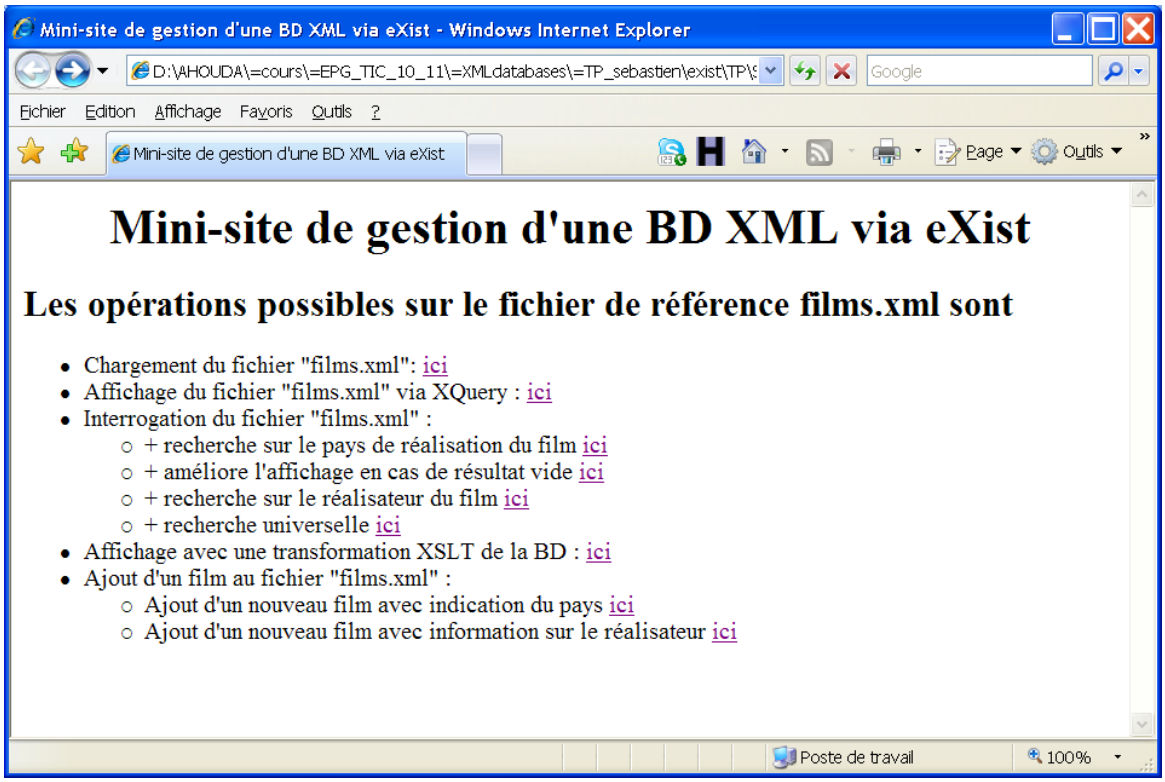
1. (Ré)initialisation de la BD
2. Affichage de la BD via XQuery
3. Interrogation de la BD
4. Affichage avec une transformation XSLT de la BD
5. Mise à jour de la BD

Pour chacune d'elles :

- testez-la
- lisez le code source des pages correspondantes
- expliquez brièvement les liens (appels) entre les pages et le contenu des pages xquery.
- sous «eXist Client Admin » :
    - Voir les modifications apparues dans le SGBD
    - Déterminer sous quel compte se connecte le site

2.3 *Extensions toujours relative au fichier films.xml*

Nous vous conseillons de dupliquer votre site actuel (les 2 parties html et query) et d'y réaliser les modifications. A la fin votre fichier index.html pourra avoir l'allure suivante :



2.3.1 Affichage via XQuery :

Modifiez l'affichage réalisé par XQuery en intégrant le pays du film et son année de production et en ne précisant pas un champ s'il est vide.

2.3.2 Requête XQuery :

1. Etendez la page de requête en autorisant une recherche sur le pays de réalisation du film.
2. Dans le fichier « query.xqy », créez une fonction qui retourne la forêt des éléments FILM qui correspondent aux critères de la requête de l'utilisateur. Utilisez alors cette fonction pour afficher le message « Aucun résultat » si cette liste est vide, sinon garder l'affichage actuel.
3. Etendez la page de requête en autorisant une recherche sur les « réalisateurs du film ».
4. Proposez une recherche plus souple: tous les éléments « contenant » le mot saisi par l'utilisateur insensible à la casse du mot.

2.3.3 Transformation XSLT :

1. Observez le résultat de la transformation XSLT, et analysez les imperfections (on regardera le code source de la page produite). Faites une sauvegarde du fichier affiche.xsl puis corrigez les imperfections en modifiant directement le fichier de transformation.

2. Afin d'éviter ces problèmes d'affichage, une autre solution est de produire directement un fichier XML contenant en en-tête l'appel à la feuille de style XSLT. Mettez en œuvre cette technique.

2.3.4 Mise à jour :

1. Ecrire une page de mise à jour avec un effet identique à l'actuelle mais en passant directement le update de Xquery et non le update du module d'extension.

2. Etendez la page de mise à jour en autorisant la définition du pays de réalisation du film.

3. Etendez la page de mise à jour en autorisant la définition du réalisateur du film. Si ce réalisateur n'existe dans la base de données, il faudra l'ajouter au sein d'un élément ARTISTE.

2.4 Extensions à la manipulation à d'autres fichiers

2.4.1 Travail à faire au préalable:

Créez une collection « MasColTP » sous « eXist » qui appartient à l'utilisateur « guest ».

2.4.2 Contenu de collection MasColTP:

Affichez un message disant que la collection est vide si elle ne contient aucun fichier sinon le noimbre de fichiers et la liste des noms de fichiers qui sont dans la collection «MasColTP ».

2.4.3 Chargement dans la collection MasColTP:

Proposez à l'utilisateur de choisir lui-même le fichier qu'il souhaite utiliser pour initialiser la base de données. Retourner lui le message explicite suivant:

Votre fichier se trouve ici: ….

Indication:

- *Modifier la page index.html*
- Regarder entre autre dans la documentation de eXist: get-parameter, ***get-uploaded-file-name***
- Utiliser la balise HTML <input> avec l'attribut type="file".

Annexe : pour les plus curieux gérer le cas du fichier vide et afficher par exemple :

Vous devez choisir un fichier si vous voulez le charger dans eXist.

Comme exemple d'utilisation: Charger le fichier films.xml dans cette collection

2.5 Utilisation de Lucene

Pour les exercices suivants, vous allez utiliser les index Lucene. Pour vous aider, vous pouvez vous référer à [S7] et écrivez les requêtes via l'une des interfaces qui vous ont été présenté dans le document « annexe – installation eXist ».

2.5.1 Utilisation des index

Ajouter à votre fichiers *films.xml* (à travers votre site un film TITRE : Mastest Policier , Annee :2011) En utilisant l'index décrit dans le document relatif au paramétrage d'index en annexe, quels sont les résultats des requêtes suivantes et pourquoi ?

1. //FILM[GENRE="Policier"]/TITRE

2. //FILM[ft:query(., 'policier')]/TITRE

3. //FILM[ft:query(GENRE, 'Policier')]/TITRE

Si vous avez suivi scrupuleusement ce qui vous a été dit de faire jusqu'à maintenant, vous devriez voir dans les réponses aux requêtes un phénomène de duplication de l'élément FILM. Expliquez ce phénomène en écrivant une requête judicieuse…

2.5.2 Ecriture de requêtes XML

En utilisant le même index, écrivez des requêtes permettant de :

1. Retrouver la liste des films contenant les termes « mourant » et/ou « sœur »

2. Retrouver la liste des films contenant les termes commençant par « cent » en utilisant

    - Les wildcard
    - Les regexp

3. Retrouver la liste des films contenant le terme « sœur », mais pas « mourant »

4. Retrouver la liste des films contenant les termes « policier » et « femme » à moins de 8 mots d'écarts.

3 Référence

[S1] XML -- TP 7 : Outil eXist (Source d'inspiration de ce TP, sauf la partie Lucene)
http://www.lif.univ-mrs.fr/~preynier/XML/tp7/tp7.html

[S2] eXist - Examples (*)
http://localhost:8080/exist/examples.xml

[S3] eXist – Xquery Function Documentation (*)
http://localhost:8080/exist/xquery/functions.xql

[S4] eXist – Developer's Guide (*)
http://localhost:8080/exist/devguide_xquery.xml

[S5] eXist – XQuery (*)
http://localhost:8080/exist/xquery.xml

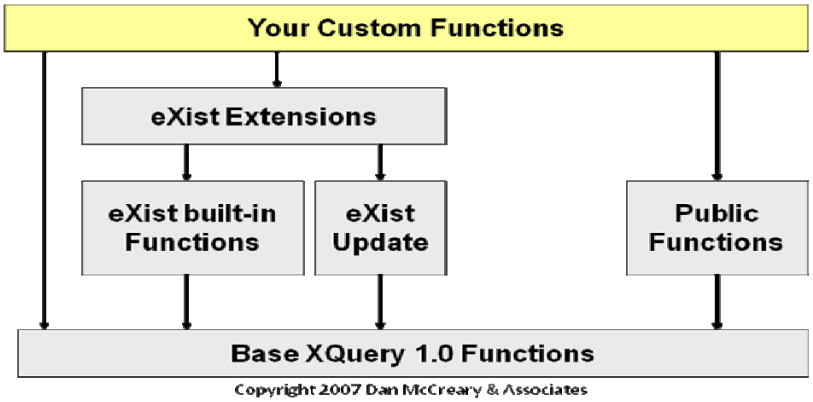
[S6] eXist – XQuery Sandbox (*)
http://localhost:8080/exist/sandbox/sandbox.xql

[S7] eXist – Lucene-based Full Text Index
http://exist-db.org/lucene.html

(*) : Disponible une fois eXist installé et démarré

4 Annexe : Fonctions utiles dans eXist

Nous utilisons dans ce TP du Xquery dans le cadre de « eXist ». Nous écrivons des fonctions Xquery qui peuvent utiliser les ~384 fonctions d'eXist:



Voici une liste non exhaustive des fonctions qui peuvent vous être utiles :

4.1 Common XQuery Built-in Functions

- functions for getting data
  - doc, collection
- numeric functions
  - abs, avg, ceiling, floor, max, min, number, round, round-half-to-even, sum
- date and time functions
  - current-dateTime, current-date, current-time, implicit-timezone
- set functions
  - distinct-values, starts-with, deep-equal, empty, subsequence
- common string functions:
  - concat, lower-case, substring, substring-before, substring-after, string-join, upper-case
- logic
  - boolean, not, false, true
- uri-functions
  - base-uri, document-uri, namespace-uri, namespace-uri-for-prefix

4.2 Common eXist Built-in Extensions

- request – processing incoming HTTP requests
  - create-session, get-parameter, get-data, get-session, get-context-path, get –cookie-names, get-cookie-value, get-header, get-hostname, get-method, get-server-port
- response – Setting outgoing HTTP responses
  - redirect, set-cookie, set-header, set-status-code
- xmldb
  - create collection, store, move, copy, change permissions, created, get-child-collections, remove, rename
- update
  - insert, update, replace, rename, delete
- session
  - manage session variables
- system
  - get system status like build number, memory-free or index data
- text – filter, fuzzy-match-all, fuzzy-match-any, highlight-matches, kwic-display, match-all, match-any
- ngram – full-text indexing contains, starts-with, ends-with, matches
- util – call, binary-doc, complie, eval, exclusive-lock, index-keys, log-system-out, random

5 Annexe : structure du site du TP

Voici une vue générale de la localisation des différents fichiers que vous manipulez dans ce TP