TP .Net

Gestion d'une bibliothèque

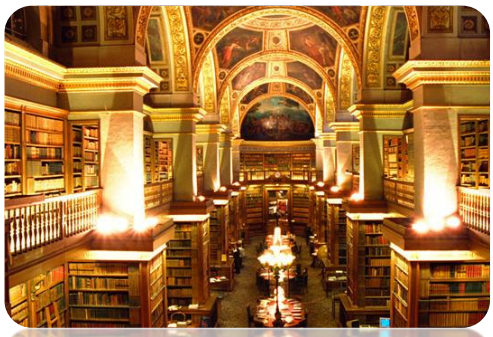
Lemettre Arnaud

Version 1.0

7 Pages

28/02/2009

EPITA-MTI2014-NET-TP-Biblio



Multimédia
Technologies de l'Information

ÉCOLE D'INGÉNIEURS EN INFORMATIQUE

Propriétés du document

Auteur	**Lemettre Arnaud**
Version	1.0
Nombre de pages	7
Références	EPITA-MTI2014-NET-TP-Biblio

Historique du document

Date de vision	Version	Auteur	Changements
27/02/2009	0.1	Lemettre	création

Site de référence

description	url
Site MTI	
Blog MTI	

Sommaire

Introduction

Le but de cette série de TP sera la gestion d'une bibliothèque. Ce tp se décompose en plusieurs parties. Toutes les semaines vous aurez une partie à rendre. Cette partie sera évaluée. Ce travail est à faire individuellement, tout code similaire sur deux personnes sera considéré comme un travail non rendu et non négociable.

Durant cette série de TP les notions suivantes seront abordées :

- modélisation BDD
- création de script SQL (Base, tables, données)
- création procédures stockées
- architecture d'une solution
- gestion des classes
- utilisation des pages aspx et composants
- gestion des XML

Chaque partie à rendre ne demande pas plus de 3h de travail par semaine, Bonne chance ;)

Contexte

Une bibliothèque municipale souhaite se moderniser et passer à l'air informatique pour gérer les livres présents dans ces rayonnages, ainsi que le système d'emprunt.

Vous aurez donc en charge de modéliser la base de données, et de faire un moteur de recherche simplifié, le système de réservation de livres, et celui d'emprunt dans une autre bibliothèque.

L'ensemble de la solution devra utiliser SQL Server 2012 pour la base de données, et le Framework 4.5 de .Net.

Le rendu de chaque lot, correspond au rendu de chaque partie.

Partie 3

Ce lot doit fournir la capacité à la solution de pouvoir faire une demande de prêt d'un livre à une autre bibliothèque. Le format d'échange pour ce type de prêt entre bibliothèques est un fichier XML.

Renseignements

Le bibliothécaire nous a fourni le schéma XML :

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Books">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="book">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="isbn" type="xs:string" />
              <xs:element name="author" type="xs:string" />
              <xs:element name="number" type="xs:unsignedByte" />
              <xs:element name="DateBack" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="name" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="IDLibrary" type="xs:unsignedInt" use="required" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Ce qui peut générer un fichier de ce genre :

```
<?xml version="1.0" encoding="utf-8" ?>
<Books IDLibrary="123445">
  <book name="Un risque Calculé">
    <isbn>2-266-15584-9</isbn>
    <author>Neville</author>
    <number>23</number>
    <DateBack>20/08/2009</DateBack>
  </book>
  <book name="Le sang du temps">
    <isbn>978-2-266-16753-6</isbn>
    <author>Chattam</author>
    <number>2</number>
    <DateBack>13/07/2009</DateBack>
  </book>
</Books>
```

Travail à faire

Réaliser une classe métier et d'accès au fichier XML qui permet de gérer les échanges de XML. Les fonctions de chargement devront prendre en paramètre le chemin d'accès des fichiers XML. Et retourner une liste d'objet pouvant stocker l'ensemble des informations d'échanges afin de pouvoir les traiter dans l'interface. En cours nous avons vu plusieurs types de parseur pour cet exercice il vous est demandé de réaliser 2 types de parseur. La première version devra être avec les XmlDocument et l'autre version avec Linq to XML.

La génération des fichiers devra correspondre au schéma XML fourni. Vous devez fournir également 2 types de génération XML. En paramètre afin de générer le fichier XML, les fonctions devront prendre une liste d'objets, correspondant aux mêmes objets que vous retournez dans vos fonctions de chargement.

Exemple :

```
List<DBO.Echange> list = BusinessManagement.Echange.LoadXML("echange.xml");
BusinessManagement.Echange.GenerateXML(list);
```

De ce fait le fichier lu et le fichier généré doit être exactement pareil.

Bonus :

- Vérification du schéma
- Signature du fichier XML avec une méthode SSL (signature enveloppée)

⚠ Attention
Si vous utilisez la signature du fichier XML, il vous faudra modifier le schéma de validation ! Sinon celui-ci ne fonctionnera plus

Modalité de rendu

Les fichiers seront à rendre dans une tarball ayant pour nom :

login_l.zip

Cette tarball devra comprendre à la racine:

- un script de création de votre base ainsi que des tables.
  Nom : CreateKBBooks.sql
- Un script pour remplir votre base avec des données
  Nom : DataKBBooks.sql
- Un dossier contenant la solution Visual Studio qui devra compiler.
  Nom : login_lKBBooks

Le tout à envoyer sur l'adresse mti.rendu.dotnet@gmail.com avec les balises suivantes :

[MTI2014][NET][login_l][Biblio] partie 3