

XML TP1

E. Bruno

26 octobre 2012

<http://isis.univ-tln.fr/~bruno/>

Table des matières

1	TP1 - Edition de documents XML bien-formés et valides	5
1.1	Objectifs	5
1.2	Les outils pour l'édition et validation de documents	5
1.3	Les recommandations	5
1.4	Validation d'un document associé à une DTD	5
1.5	Edition d'un document XML	6
1.6	Espaces de noms	6
1.7	Création d'une définition de type de documents : DTD	6
1.7.1	Baliser des documents	6
1.7.2	Construire une DTD	6
1.8	Une DTD industrielle, Docbook	6
1.8.1	Créer un document conforme à la DTD docbook	7
1.8.2	Transformer un document XML	7

<http://isis.univ-tln.fr/brunol/>

<http://isis.univ-tln.fr/~bruno/>

Chapitre 1

TP1 - Edition de documents XML bien-formés et valides

1.1 Objectifs

- Utilisation des outils de base pour la création et l'édition de documents XML ¹.
- Vérification des documents bien formés et validation d'un document par rapport à une DTD ².
- (Optionnel) Exemple de transformation avec XSLT ³ vers HTML ⁴, XSL ⁵ puis vers pdf.

1.2 Les outils pour l'édition et validation de documents

Les outils suivant sont utilisables pour l'édition :

- l'IDE eclipse
- L'éditeur Emacs <http://www.gnu.org/software/emacs/emacs.html>
 - Le mode psxml <http://sourceforge.net/projects/psxml/>, avec le mode nxml et le mode ttdt <http://www.menteith.com/ttdt/>
- Le logiciel commercial xmlspy (<http://www.xmlspy.com>) disponible sous windows ou sous linux avec wine.
- Le logiciel commercial oxygen (<http://www.oxygenxml.com>) sur `sis /usr/local/Oxygen`

Dans le cadre de ces TP, nous utiliserons

- eclipse comme interface d'édition <http://eclipse.org/juno/>
- Xerces2 version java (<http://xerces.apache.org/>) pour la validation en ligne de commande
- Saxon pour Java (<http://saxon.sourceforge.net/#F9.4HE>) comme processeur XSLT³ pour la transformation de document XML¹ (Saxon est aussi un parseur XML¹ et un processeur XQuery).
- Il faudra donc une machine virtuelle java installée JDK⁶ SE (au moins version 6) (<http://www.oracle.com/technetwork/java/jav>

1.3 Les recommandations

- XML¹ : <http://www.w3.org/TR/xml11>
- Namespaces : <http://www.w3.org/TR/xml-names11>

1.4 Validation d'un document associé à une DTD

Le fichier books.dtd.zip décrit un modèle de document qui permet de représenter une collection de livres. Le fichier books-errors.xml est un document XML¹ exemple qui est une instance de la dtd précédente, il contient des erreurs. Deux types d'erreurs peuvent apparaître. Des erreurs de non-conformité avec le langage XML¹ (balises

1. Extensible Markup Language
2. Document Type Definition
3. Extensible Stylesheet Language Transformations
4. HyperText Markup Language
5. Extensible Stylesheet Language
6. Java Development Kit

non fermés, ...), on dira alors que le document n'est pas bien formé ; et des erreurs de non conformité par rapport à une DTD², on dira alors que le document n'est pas valide par rapport à cette DTD².

Pour vérifier la conformité d'un document à la norme XML¹, on utilise un éditeur XML¹ (eclipse pour nous) ou un parser (Xerces en ligne de commande) :

```
java -cp xerces-2_11_0/xercesImpl.jar:xerces-2_11_0/xercesSamples.jar:xerces-2_11_0/xml-apis.jar
dom.Counter -v books-errors.xml
```

En fait, il s'agit d'une application java qui analyse le document en comptant les éléments mais au et à mesure elle vérifie le document.

Corriger les erreurs syntaxiques du document `books-errors.xml` pour le rendre bien formé.

Pour valider un document XML¹ par rapport à une DTD², on ajoute un DOCTYPE dans le document pour le lier à cette DTD² (enlever le commentaire autour de celui qui est présent et modifier le chemin). Le parser l'utilisera pour vérifier la validité du document. Rendre le document `books-errors.xml` valide par rapport à la dtd `books.dtd`.

```
java -cp xerces-2_11_0/xercesImpl.jar:xerces-2_11_0/xercesSamples.jar:xerces-2_11_0/xml-apis.jar
dom.Counter -v books.xml
```

1.5 Edition d'un document XML

En utilisant eclipse, exercez vous à ajouter des éléments, des attributs et du texte en insérant deux ou trois nouveau livre dans le document XML¹.

Vous mettrez éventuellement la DTD² à jour, et vous validerez le document modifié.

1.6 Espaces de noms

Modifier le document précédent en utilisant les espaces de noms. Vous structurerez de façon plus fine les auteurs pour leur ajouter un `title` (Dr., Pr., Ms., Mrs., Miss). Pour distinguer les éléments ayant le même nom, vous définirez deux espaces de noms, un pour les livres et un pour les auteurs.

DTD²

1.7 Création d'une définition de type de documents : DTD

1.7.1 Baliser des documents

Proposer une structuration XML¹ pour les classes et interfaces de la javadoc <http://java.sun.com/j2se/1.5.0/docs/api/index.html>. Vous détaillerez un exemple avec les classes `String` et `StringBuilder`. Attention, le site Web est orienté présentation mais votre document XML¹ devra décrire la structure logique. Vous utiliserez en particulier la possibilité de décrire des listes et des tableaux et les références internes.

1.7.2 Construire une DTD

A partir des balisages que vous venez de définir, proposer une DTD² `javadoc.dtd` et valider les deux documents.

1.8 Une DTD industrielle, Docbook

La DTD² docbook (ainsi qu'une version simplifiée) sont présentés sur les sites <http://www.docbook.org> et <http://www.oasis-open.org/docbook/>. Un exemple simple de document se trouve dans le fichier `sample-docbook.xml`.

La documentation complète de docbook écrite en XML¹ se trouve ici : <http://docbook.svn.sourceforge.net/viewvc/docbook/trunk/> une version HTML⁴ (générée automatiquement) est disponible ici : <http://www.docbook.org/tdg/en/html/docbook.html>.

La DTD² est ici <http://www.oasis-open.org/docbook/xml/>

les feuilles de style là : <http://sourceforge.net/projects/docbook/> (Plus d'information <http://sagehill.net/docbookxsl/index.html>).

1.8.1 Créer un document conforme à la DTD docbook

Écrire le manuel d'utilisation d'un projet que vous avez réalisé en utilisant la DTD² docbook.

1.8.2 Transformer un document XML

Un document XML¹ peut être transformé en un autre document XML¹ (ou en un fichier texte) en utilisant des feuilles de style XSLT³ (celles-ci seront étudiées ultérieurement en cours, nous allons ici simplement les utiliser).

Trois feuilles de style sont proposées pour transformer un document XML¹ valide pour la DTD² docbook en

- HTML⁴ : docbook-xsl-1.77.1/html/docbook.xsl
- XHTML⁷ : docbook-xsl-1.77.1/xhtml/docbook.xsl
- XSL⁵-FO : docbook-xsl-1.77.1/fo/docbook.xsl

Il faut utiliser un processeur XSLT³ (Saxon par exemple). La commande pour appliquer une feuille de style de xhtml est :

```
java -cp saxon9he.jar net.sf.saxon.Transform -strip:all -s:sample-docbook.xml  
-xsl:docbook-xsl-1.77.1/xhtml/docbook.xsl -o:sample-document.xhtml
```

Transformer le document de l'exercice précédent en un document HTML⁴, puis en un document XHTML⁷/XHTML5.

La troisième feuille de style produit un document XML¹ dans un format appelé FO (Formating Object) qui décrit un document formaté. Ce document peut ensuite être converti vers un format propriétaire (dvi, ps, pdf, ...).

Pour réaliser cette transformation, divers outils sont disponibles par exemple FOP (<http://xml.apache.org/fop/>)

```
java -cp saxon9he.jar net.sf.saxon.Transform -strip:all -s:sample-docbook.xml  
-xsl:docbook-xsl-1.77.1/fo/docbook.xsl -o:sample-document-fo.xml  
fop -fo sample-document-fo.xml -pdf sample-document-fo.pdf
```