Université Abderrahmane Mira de Bejaia
Faculté des Sciences Exactes
Département d'informatique
Année Universitaire : 2017/2018

2ème Année Master GL – (S3) Module : **Services Web**

TP 02 : Manipulation des documents XML

Objectif

Le but de ce TP consiste à valider des documents XML et manipuler des fichiers XML avec l'API DOM en langage Java.

Rappel :

DOM ou **Document Object Model**, son nom complet, est ce qu'on appelle un **parseur XML**, c'est-à-dire, une technologie grâce à laquelle il est possible de lire un document XML et d'en extraire différentes informations (éléments, attributs, commentaires, etc.) afin de les exploiter.

Comme pour la plupart des technologies web, DOM est un standard du W3C et ce, depuis sa première version en 1998. Actuellement, cette technologie en est à sa troisième version.

Il est très important de noter que DOM est une recommandation complètement *indépendante* de toute plate-forme et langage de programmation. Au travers de DOM, le W3C fournit une recommandation, c'est-à-dire une manière d'exploiter les documents XML.

Aujourd'hui, la plupart des langages de programmation propose leur implémentation de DOM : C, C ++, Java, C#, Perl, PHP, etc.

Activité 01 : Correction et Validation de documents XML

En utilisant l'outil xmlvalidator, vérifier les documents XML suivants avec leurs fichiers respectifs (DTD ou XML schema).

Fichier : wg.xml

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE gdt SYSTEM "wg.dtd">
<gdt month="February" year="2004">
  <expose day="5">
    <speaker>Nicolas Baudru</speaker>
    <title>Netcharts et HMSC</title>
    <time>12h45</time>
    <salle>102</salle>
  </expose>
  <expose day="12">
    <comment>Exposé proposé par François Denis, équipe BDA
</comment>
    <speaker>Liva Ralaivola</speaker>
    <title>apprentissage statistique, noyaux et
applications à la
    bio-informatique</title>
    <time>14h</time>
    <salle>104</salle>
  </expose>
  <expose day="19">
    <comment>pas de groupe de travail</comment>
  </expose>
   <expose day="26">
    <speaker>Pedro d'Argenio</speaker>
    <title>Secure Information Flow by Self-Composition</title>
    <time>12h45</time>
    <salle>102</salle>
  </expose>
</gdt>
```

Fichier : wg.dtd

```
<!ELEMENT gdt  (expose*)>
<!ATTLIST gdt
        month CDATA #REQUIRED
        year CDATA #REQUIRED>
<!ELEMENT expose  (comment?,
(speaker, title, time,salle)?)>
<!ATTLIST expose day CDATA
#REQUIRED>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT speaker(#PCDATA)>
<!ELEMENT title       (#PCDATA)>
<!ELEMENT time       (#PCDATA)>
<!ELEMENT salle       (#PCDATA)>
```

Fichier : contacts.xml

```
<Contacts>
<Person>
<Firstname>John</Firstname>
<Lastname>Smith</Lastname>
<Birthday>1965-03-29</Birthday>
<Company>IBM</Company>
<Position>CEO</Position>
<Email>jsmith@ibm.com</Email>
<Email>jsmith@yahoo.com</Email>
<Address type="home">
<Company/>
<Street>23 Main St</Street>
<City>Dublin</City>
<Postcode>4</Postcode>
<Country/>
</Address>
<Address type="work">
<Street>1234 High St</Street>
<City>London</City>
<ZIP>1234</ZIP>
<Country>Ireland</Country>
</Address>
</Person>
<Person>
<Firstname>Tom</Firstname>
<Lastname>Dunne</Lastname>
<Company>Today FM</Company>
<Position/>
<Email>tom.dunn@todayfm.com</Email>
</Person>
</Contacts>
```

Fichier : contacts.xsd

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="Contacts">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="Person" maxOccurs="unbounded" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="Person">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Firstname" type="xsd:string"/>
<xsd:element name="Lastname" type="xsd:string"/>
<xsd:element name="Birthday" type="xsd:date" minOccurs="0"/>
<xsd:element name="Company" type="xsd:string" minOccurs="0"/>
<xsd:element name="Position" type="xsd:string" minOccurs="0"/>
<xsd:element name="Email" type="xsd:string" minOccurs="0" maxOccurs="10"/>
<xsd:element ref="Address" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="Address">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Company" type="xsd:string" minOccurs="0"/>
<xsd:element name="Department" type="xsd:string" minOccurs="0"/>
<xsd:element name="Street" type="xsd:string"/>
<xsd:element name="City" type="xsd:string"/>
<xsd:choice>
<xsd:element name="Postcode" type="xsd:string"/>
<xsd:element name="ZIP" type="xsd:string"/>
</xsd:choice>
<xsd:element name="Country" type="xsd:string"/>
</xsd:sequence>
<xsd:attribute name="type" type="xsd:string" use="required"/>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

<u>Activité 02</u> : Afficher un fichier XML (utilisation de l'API jdom)

Ce programme permet de lire un fichier XML et de récupérer les informations (Parser XML).
On vous donne le fichier à parser (repertoire.xml) :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<repertoire>
    <!-- John DOE -->
    <personne sexe="masculin">
        <nom>DOE</nom>
        <prenom>John</prenom>
        <telephones>
            <telephone type="fixe">01 02 03 04 05</telephone>
            <telephone type="portable">06 07 08 09 10</telephone>
        </telephones>
    </personne>
</repertoire>
```

Et le fichier java correspondant :

```
package ecrirefichierxml;
import java.io.File;
import java.io.IOException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
public class affxml {
    public static void main(final String[] args) {
        /*
         * Etape 1 : récupération d'une instance de la classe "DocumentBuilderFactory"
         */
        final DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

        try {
            /*
             * Etape 2 : création d'un parseur
             */
            final DocumentBuilder builder = factory.newDocumentBuilder();

           /*
            * Etape 3 : création d'un Document
            */
           final Document document= builder.parse(new File("repertoire.xml"));

           //Affiche du prologue
           System.out.println("*************PROLOGUE************");
           System.out.println("version : " + document.getXmlVersion());
           System.out.println("encodage : " + document.getXmlEncoding());
            System.out.println("standalone : " + document.getXmlStandalone());

           /*
            * Etape 4 : récupération de l'Element racine
            */
           final Element racine = document.getDocumentElement();

           //Affichage de l'élément racine
           System.out.println("\n*************RACINE************");
           System.out.println(racine.getNodeName());

           /*
            * Etape 5 : récupération des personnes
            */
           final NodeList racineNoeuds = racine.getChildNodes();
           final int nbRacineNoeuds = racineNoeuds.getLength();

           for (int i = 0; i<nbRacineNoeuds; i++) {
               if(racineNoeuds.item(i).getNodeType() == Node.ELEMENT_NODE) {
```

```
                    final Element personne = (Element) racineNoeuds.item(i);

                //Affichage d'une personne
                System.out.println("\n*************PERSONNE************");
                System.out.println("sexe : " + personne.getAttribute("sexe"));

                /*
                 * Etape 6 : récupération du nom et du prénom
                 */
                final Element nom = (Element) personne.getElementsByTagName("nom").item(0);
                final Element prenom = (Element)
personne.getElementsByTagName("prenom").item(0);

                //Affichage du nom et du prénom
                System.out.println("nom : " + nom.getTextContent());
                System.out.println("prénom : " + prenom.getTextContent());

                /*
                 * Etape 7 : récupération des numéros de téléphone
                 */
                final NodeList telephones = personne.getElementsByTagName("telephone");
                final int nbTelephonesElements = telephones.getLength();

                for(int j = 0; j<nbTelephonesElements; j++) {
                    final Element telephone = (Element) telephones.item(j);

                       //Affichage du téléphone
                       System.out.println(telephone.getAttribute("type") + " : " +
telephone.getTextContent());
                }
            }
          }
        }
        catch (final ParserConfigurationException e) {
            e.printStackTrace();
        }
        catch (final SAXException e) {
            e.printStackTrace();
        }
        catch (final IOException e) {
            e.printStackTrace();
        }
    }
}
```

<u>Activité 03</u> : Créer un fichier XML (utilisation de l'API jdom)

1. Créer un nouveau projet de type *Java Application* nommé *EcrireFichierXML*.
2. Copier et exécuter le code suivant :

```
package ecrirefichierxml;
import java.io.*;
import org.jdom.*;
import org.jdom.output.*;
 class Main
{ static Element racine = new Element("personnes");
static org.jdom.Document document = new Document(racine);
public static void main(String[] args)
{ Element etudiant = new Element("etudiant");
racine.addContent(etudiant);
Attribute classe = new Attribute("classe", "P2");
etudiant.setAttribute(classe);
Element nom = new Element("nom");
nom.setText("Nom1");
etudiant.addContent(nom);
try { XMLOutputter sortie = new XMLOutputter(Format.getPrettyFormat());
sortie.output(document, new FileOutputStream("./Exemple1.xml")); }
catch (java.io.IOException e) { }
} }
```

3. Analyser le code ci-dessus et écrire un programme qui vous permet de générer le fichier XML suivant :

```
<?xml version="1.0"?>
<class>
   <student rollno="393">
      <firstname>dinkar</firstname>
      <lastname>kad</lastname>
      <nickname>dinkar</nickname>
      <marks>85</marks>
   </student>
   <student rollno="493">
      <firstname>Vaneet</firstname>
      <lastname>Gupta</lastname>
      <nickname>vinni</nickname>
      <marks>95</marks>
   </student>
   <student rollno="593">
      <firstname>jasvir</firstname>
      <lastname>singn</lastname>
      <nickname>jazz</nickname>
      <marks>90</marks>
   </student>
</class>
```