



## ESISAR

### NE449 – Programmation répartie

### TDM Bonus2

#### Table des matières

1 Objectifs du TDM Bonus 2.....	1
2 Exercice 1 : Découverte de XML - Parsing d'un fichier XML.....	1
3 Exercice 2 : Mise en œuvre d'un service Web XML.....	3
4 Exercice 3 : Mise en œuvre d'un service Web XML – Suite.....	4
5 Exercice 4 : Mise en œuvre de plusieurs services Web.....	4
6 Exercice 5 : Mise en œuvre d'un service Web JSON.....	4
7 Exercice 6 : Pour aller beaucoup plus loin .....	4

Ce TDM BONUS2 comprend un ensemble d'exercices optionnels pour les étudiants qui souhaitent aller plus loin.

#### 1 Objectifs du TDM Bonus 2

Ce TDM est optionnel. Il va permettre de découvrir les web services. Ce TP sera réalisé en JAVA. Dans ce TP, vous réaliserez uniquement la partie cliente, **pour la partie serveur vous devrez rechercher et utiliser des Web Services disponibles librement sur Internet.**

#### 2 Exercice 1 : Découverte de XML - Parsing d'un fichier XML

XML est une norme développée par le W3C, et signifie eXtensible Markup Language.

XML n'est pas un langage mais un méta-langage. Ce méta-langage permet de concevoir de nouveaux langages (eXtensible) à balises (Markup), comme par exemple XHTML, SVG, ...

Auto formez vous à XML avec les références suivantes :

<http://www.ibm.com/developerworks/library/x-newxml/>

<http://www.ibm.com/developerworks/xml/tutorials/xmlintro/section2.html>

Xpath permet de récupérer facilement des informations depuis un fichier XML.

Auto formez vous à X Path avec la référence suivante:

<http://jean-luc.massat.perso.luminy.univmed.fr/ens/xml/4-xpath.html>

Le langage Java contient par défaut les outils nécessaires pour récupérer des informations dans des fichiers XML avec Xpath. Voici un exemple :

```
import java.io.IOException;
import javax.xml.parsers.*;
import javax.xml.xpath.*;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class ParserXPath
{
    public ParserXPath()
    {
    }

    public void execute() throws ParserConfigurationException, SAXException,
    IOException, XPathExpressionException
    {
        DocumentBuilderFactory domFactory = DocumentBuilderFactory.newInstance();
        domFactory.setNamespaceAware(true);
        DocumentBuilder builder = domFactory.newDocumentBuilder();

        Document doc = builder.parse("annuaire.xml");
        XPath xpath = XPathFactory.newInstance().newXPath();

        //
        System.out.println("Liste des prenom");
        XPathExpression expr = xpath.compile("//personne/prenom/text()");
        Object result = expr.evaluate(doc, XPathConstants.NODESET);
        NodeList nodes = (NodeList) result;
        for (int i = 0; i < nodes.getLength(); i++)
        {
            System.out.println(nodes.item(i).getNodeValue());
        }

        System.out.println("Liste des ages");
        expr = xpath.compile("//personne/age/text()");
        result = expr.evaluate(doc, XPathConstants.NODESET);
        nodes = (NodeList) result;
        for (int i = 0; i < nodes.getLength(); i++)
        {
            System.out.println(nodes.item(i).getNodeValue());
        }
    }

    public static void main(String[] args) throws Exception
    {
    }
}
```

```
{  
    ParserXPath xpath = new ParserXPath();  
    xpath.execute();  
}
```

### Contenu du fichier annuaire.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<annuaire>  
  <personne>  
    <nom>A</nom>  
    <prenom>Alice</prenom>  
    <age>22</age>  
  </personne>  
  
  <personne>  
    <nom>B</nom>  
    <prenom>Bob</prenom>  
    <age>23</age>  
  </personne>  
  
  <personne>  
    <nom>C</nom>  
    <prenom>Clara</prenom>  
    <age>24</age>  
  </personne>  
</annuaire>
```

Mettez en œuvre cet exemple.

## 3 Exercice 2 : Mise en œuvre d'un service Web XML

Réalisez en programme qui prenne en entrée :

- le nom d'une ville de départ
- le nom d'une ville d'arrivée

et qui affiche en résultat la distance en km entre les deux villes.

Votre programme s'appuiera sur un Web Service fournissant ces données en XML.

**Pour cela, vous devez utiliser une librairie Java prête à l'emploi permettant le téléchargement d'une page (mot clé : httpclient).**

## 4 Exercice 3 : Mise en œuvre d'un service Web XML – Suite

Faites évoluer votre programme précédent pour afficher maintenant le parcours entre deux villes. Le parcours sera une liste de points, chaque point est représenté par sa latitude et sa longitude.

## 5 Exercice 4 : Mise en œuvre de plusieurs services Web

Faites évoluer votre programme précédent pour afficher maintenant l'altitude maximum qui est franchie lors du parcours entre deux villes.

Bonus : afficher une photo de l'endroit de ce point d'altitude maxi.

## 6 Exercice 5 : Mise en œuvre d'un service Web JSON

JSON est un format concurrent (ou complémentaire) à XML. Lisez la RFC associée :

<http://tools.ietf.org/html/rfc4627>

Modifier un exercice précédent pour le faire en JSON.

## 7 Exercice 6 : Pour aller beaucoup plus loin ...

Faites évoluer le programme de l'exercice 4 pour afficher maintenant le film du trajet entre deux villes. Votre programme permettra de saisir le lieu de départ, le lieu d'arrivée, et après un temps de chargement l'utilisateur pourra visionner le film du trajet entre ces deux points.

Vous trouverez dans le lien suivant une vidéo réalisée avec un tel logiciel :

<http://chamilo2.grenet.fr/inp/courses/ESISAR4AMNE441/document/TP/drive.avi>