

## TP N° 02

### Objectifs

L'objectif de cette série de TP est de :

- L'objectif principal de cette série est la création d'un document XML à travers le code. Contrairement au "parsing" qui peut être effectué de plusieurs manières, la création d'un document nécessite la manipulation du DOM. L'objectif de cette activité est de donner un exemple de création d'un document XML en Java.

### Etapes à suivre

La création d'un document Xml en utilisant l'API DOM sous JAVA ressemble au parsing, mais, au lieu de "parse", on crée un nouveau document par "newDocument", ce qui nous donne le code suivant :

```
DocumentBuilderFactory builderFactory =  
    DocumentBuilderFactory.newInstance();  
DocumentBuilder builder = dbFactory.newDocumentBuilder();  
Document document = builder.newDocument();
```

Pour créer les différents éléments :

- document.createElement() : permet de créer un élément,
- document.createAttribute() : permet de créer un attribut,
- document.createTextNode() : permet de créer un nœud text (contenu),

La définition de la valeur de l'attribut est séparée de la création

- attribut.setValue()

Ensuite, il faut construire le DOM :

- element.appendChild() : ajouter un élément fils à "element",
- element.setAttributeNode() : attacher un attribut à "element".

**Note importante** : il faut créer un premier Element (nœud) racine et l'attacher au document créé :

- document.appendChild()

Finalement, le document créé est envoyé vers un fichier. Pour cela un ensemble d'objets doit être créé :

- Définir le document comme un flux source,
- Définir un flux de destination qui va recevoir le contenu de la source,

- Définir un objet de transformation qui transforme la source vers la destination.

Ces objets sont :

- La source :

```
DOMSource source = new DOMSource(doc);
```

- La destination :

```
StreamResult dest = new StreamResult(new File("fichier.xml"));
```

- Le transformateur :

```
TransformerFactory transformerFactory =
    TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
```

- La transformation :

```
transformer.transform(source, dest);
```

Il est possible d'afficher le contenu du fichier si on utilise la console comme destination :

```
StreamResult dest = new StreamResult(System.out);
```

## ***Exemple***

Soit le code suivant :

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;

public class Main {

    public static void main(String[] args) {
        try {
            DocumentBuilderFactory factory
                = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
```

```

Document document = builder.newDocument();

Element racine = document.createElement("message");
document.appendChild(racine);

Element contenu_message = document.createElement("contenu");
Node contenu_message_text
    = document.createTextNode("Bonjour tout le monde");

contenu_message.appendChild(contenu_message_text);

// Ou bien, d'une manière plus simple
Element date_message = document.createElement("date");
date_message.appendChild(document.createTextNode("01-01-
1970"));

racine.appendChild(contenu_message);
racine.appendChild(date_message);

TransformerFactory tf = TransformerFactory.newInstance();
Transformer transformer = tf.newTransformer();
DOMSource source = new DOMSource(document);

StreamResult destination = new StreamResult(System.out);
transformer.transform(source, destination);

} catch (ParserConfigurationException ex) {
    System.out.println("Erreur " + ex.getMessage());
} catch (TransformerConfigurationException ex) {
    System.out.println("Erreur " + ex.getMessage());
} catch (TransformerException ex) {
    System.out.println("Erreur " + ex.getMessage());
}
}
}

```

Ce code permet de créer le fichier XML suivant (après décoration) :

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<message>
    <contenu>Bonjour tout le monde</contenu>
    <date>01-01-1970</date>
</message>

```

## ***Travail demandé***

Soit le document XML suivant :

```
<livre>
  <titre>Java</titre>
  <prix>1200,00DA</prix>
  <auteurs>
    <auteur>J. Gosling</auteur>
  </auteurs>
</livre>
```

Ecrivez le code Java qui permet de le créer et de le sauvegarder sous le nom “exemple\_02.xml”.