TP N° 04
JAXB

Objectif

Pour simplifier la manipulation des éléments de base (comme un fichier XML), l'écosystème Java propose des frameworks, API, ou blibliothèque pour simplifier la tâche aux développeurs. Ces frameworks reposent sur les bonnes pratiques et leur manipulation peut nécessiter un certain niveau d'expertise, mais, une fois maîtrisées, ces frameworks peuvent permettre au développeur d'augmenter sa productivité et de créer des applications plus professionnelles, lisibles et faciles à maintenir.

JAXB

Présentation

"Java Architecture for XML Binding (JAXB) provides a fast and convenient way to bind XML schemas and Java representations, making it easy for Java developers to incorporate XML data and processing functions in Java applications. As part of this process, JAXB provides methods for unmarshalling (reading) XML instance documents into Java content trees, and then marshalling (writing) Java content trees back into XML instance documents. JAXB also provides a way to generate XML schema from Java objects." [1]

Oracle Java Documentation

Java Architecture for XML Binding (JAXB) offre un moyen rapide et pratique de lier des schémas XML et des représentations Java, ce qui permet aux développeurs Java d'intégrer facilement des données XML et des fonctions de traitement dans des applications Java.

Marshaller et Unmarshaller

JAXB repose sur deux objets pour lire et écrire des fichiers XML :

- Marshaller : un objet de cette classe permet de créer un fichier XML à partir d'un objet (POJO),
- Unmarshaller : un objet de cette classe permet de lire un fichier XML à partir duquel il crée un objet java (POJO).

Les deux objets sont créés à partir d'un contexte, ce dernier est créé à partir de la classe qui constitue la racine du fichier XML à lire ou à écrire.

Lier les attributs Java aux éléments et attributs XML

Pour assurer le lien entre les attributs Java et les éléments et attributs XML correspondants, JAXB fait appel à un ensemble d'annotations. Nous nous limitons à la liste suivante :

- @XmlRootElement : Définit l'élément racine XML. Les classes Java racine doivent être enregistrées avec le contexte JAXB lors de leur création.

1 https://docs.oracle.com/javase/tutorial/jaxb/intro/index.html

- @XmlElement : Associe un champ ou une propriété à un élément XML.
- @XmlElementWrapper : Mappe une collection Java à une collection enveloppée de XML.
- @XmlAttribute : Associe un champ ou une propriété à un attribut XML.
- @XmlTransient : L'élément sera ignoré et ne figurera pas dans le document XML.
- @XmlValue : Associe un champ ou une propriété à la valeur de texte d'une balise XML.

Il est possible de définir une méthode d'accès pour tout l'objet en utilisant l'annotation @XmlAccessorType en passant l'une des valeurs suivantes :

- XmlAccessType.FIELD : Chaque champ non statique de la classe liée à JAXB sera automatiquement lié à XML.
- XmlAccessType.PROPERTY : Chaque paire getter/setter dans une classe liée à JAXB sera automatiquement liée à XML.
- XmlAccessType.PUBLIC_MEMBER : Chaque paire getter/setter **publique** et chaque champ public sera automatiquement lié à XML.

Exemple

Un exemple de POJO annoté (il est possible d'ajouter le constructeur et les setter/getter) :

```
@XmlRootElement(name = "livre")
@XmlAccessorType(XmlAccessType.FIELD)
public class Livre {

    @XmlAttribute
    int id;

    @XmlElement
    String titre;

    @XmlElement
    float prix;

    @XmlElementWrapper(name = "auteurs")
    @XmlElement(name = "auteur")
    List<String> auteur;

}
```

un exemple d'un code qui permet de créer un fichier XML à partir d'une instance de cette classe :

```
List<String> auteurs = Arrays.asList("J. Gosling");
Livre livre = new Livre(1, "Java", 1200, auteurs);

JAXBContext context = JAXBContext.newInstance(Livre.class);
Marshaller m= context.createMarshaller();
m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE);
m.marshal(livre, new File("livre.xml"));
```

Le fichier XML créé sera le suivant :

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<livre id="1">
    <titre>Java</titre>
    <prix>1200.0</prix>
    <auteurs>
        <auteur>J. Gosling</auteur>
    </auteurs>
</livre>
```

Travail demandé

Il est demandé de créer une application (fenêtre Swing) qui permet de :

- Visualiser la liste des clients (d'une entreprise quelconque) dans une JTable,
- Exporter les données sous forme d'un fichier XML.