

# JAXB – JAXWS

- **JAXB**

## ***Qu'est-ce que c'est ?***

JAXB (*Java Architecture for XML Binding*) est une API qui permet de faire correspondre un document XML à un ensemble de classes et inversement par l'intermédiaire d'opérations appelées sérialisation et désérialisation (marshalling/unmarshalling). Il permet également à un développeur de facilement manipuler un document XML sans avoir à connaître le langage XML. Il existe actuellement JAXB 1.0 et JAXB 2.0. Il se différencie de son prédécesseur par ses caractéristiques suivantes :

- support uniquement des schémas XML (les DTD ne sont plus supportées)
- mise en oeuvre des annotations
- assure la correspondance bidirectionnelle entre un schéma XML et le bean correspondant.
- l'utilisation de fonctionnalités proposées par Java 5 notamment les generics et les énumérations
- le nombre d'entités générées est moins important : JAXB 2.0 génère une classe pour chaque complexType du schéma alors que JAXB 1.0 génère une interface et une classe qui implémente cette interface. Une méthode de la classe ObjectFactory est générée pour renvoyer une instance de cette classe.

## ***Comment ça fonctionne ?***

Pour modifier un document XML en objet Java il faut utiliser des annotations.

Il est nécessaire dans un premier temps il obtenir un objet du type JAXBContext en utilisant la méthode newInstance(). Cette méthode demande en paramètre une chaîne de caractères indiquant le package des interfaces générées à partir du schéma.

La méthode createMarshaller() permet d'obtenir un objet de type Marshaller. C'est cet objet qui va formater le document XML.

Il est possible de lui préciser des propriétés pour effectuer sa tâche en utilisant la méthode setProperty(). Des constantes sont définies pour ces propriétés dont les principales sont :

JAXB\_ENCODING : permet de préciser le jeu de caractères d'encodage du document XML sous la forme d'une chaîne de caractères

JAXB\_FORMATTED\_OUTPUT : booléen qui indique si le document XML doit être formaté

Les propriétés doivent être des objets.

L'appel de la méthode marshal() formate le document dont l'objet racine est fourni en premier paramètre. Il existe plusieurs surcharges de cette méthode pour préciser où est envoyé le résultat de la génération. Le second paramètre permet de préciser cette cible : un flux en sortie, un arbre DOM, des événements SAX.

### Quelques annotations :

XmlAccessorType	Ordonner les champs et propriétés dans la classe
XmlAccessorType	Préciser comment un champ ou une propriété est accédé par JAXB. Par défaut, tous les champs publics ou annotés sont pris en compte sauf ceux marqués avec @XmlTransient. Sa valeur est de type XmlAccessType qui est une énumération qui peut prendre les valeurs NONE, FIELD, PROPERTY, PUBLIC_MEMBER
XmlAnyElement	Convertir une collection d'éléments en une collection de type List<Element>
XmlAttribute	Convertir une propriété en un attribut dans le document XML
XmlElement	Convertir une propriété en un élément dans le document XML
XmlElementDecl	Associer une fabrique à un élément XML
XmlElement	Contenir plusieurs annotations @XmlElement
XmlElementWrapper	Créer un élément père dans le document XML pour des collections d'éléments
XmlEnum	Définir la façon dont une énumération est convertie dans le document XML
XmlEnumValue	Définir la valeur numérique d'un élément d'une énumération
XmlID	Convertir une propriété en un XML ID dans le document XML. Ne peut être utilisé que sur une seule propriété.
XmlIDREF	Convertir une propriété en un XML IDREF dans le document XML
XmlList	Préciser que les éléments de liste sont représentés sous la forme d'une chaîne de caractères dans laquelle chaque valeur est séparée par un espace
XmlNs	Associer un préfixe d'un espace de nommage à un URI
XmlRegistry	Marquer une classe comme possédant une ou des méthodes annotées avec @XmlElementDecl
XmlRootElement	Associer une classe ou une énumération à un élément racine XML
XmlSchema	Associer un ou plusieurs espaces de nommage à un package
XmlSchemaType	Associer un type Java ou une énumération à un type défini dans un schéma
XmlTransient	Marquer une entité ne devant pas être mappée dans le document XML
XmlType	Associer une classe ou une énumération à un type d'un schéma XML. L'attribut propOrder permet de définir l'ordre des champs dans le document XML
XmlValue	Convertir une classe en une valeur simple

### Exemple :

Nous avons créé une classe Personne avec des attributs basiques comme le nom, le prénom et la date de naissance, puis les méthodes getters et setters et les constructeurs.

Nous n'oublions pas de mettre un « import java.io.Serializable »

Dans une classe appelée Sérialisation, on écrit le code suivant nous permettant d'instancier une nouvelle JAXBContexte ainsi qu'une personne et ses attributs. De plus nous retrouverons la conversion dans un fichier « resultat.xml »

```
try {  
    JAXBContext jc = JAXBContext.newInstance(Personne.class);  
    Personne p_1 = new Personne("Souf", "Souf", new Date(31,1,1999));  
    Marshaller m = jc.createMarshaller();  
    m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);  
    m.marshal(p_1, new File("resultat.xml"));  
  
    System.out.println("END");  
} catch (JAXBException e) {  
    e.printStackTrace();  
}
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<personne>  
  <date>  
    <a>1999</a>  
    <j>31</j>  
    <m>1</m>  
  </date>  
  <nom>Souf</nom>  
  <prenom>Souf</prenom>  
</personne>
```

On constate que les 3 attributs sont présents.

- **JAX-WS**

*Qu'est-ce que c'est ?*

JAX-WS (Java Architecture for XML Web Services) est une API permettant de créer et d'utiliser des services Web SOAP (Simple Object Access Protocol) qui est spécification XML permettant d'envoyer des messages sur un réseau. L'objectif de cet API est de faciliter le développement des Web services et de leurs clients consommateurs avec le minimum de code.