

Atelier

Génération d'un document XML, XSD et java via JAXB (Marshalling et Unmarshalling)

Objectifs

Générer un fichier XSD à partir d'une classe java

Générer une classe java à partir d'un fichier XSD

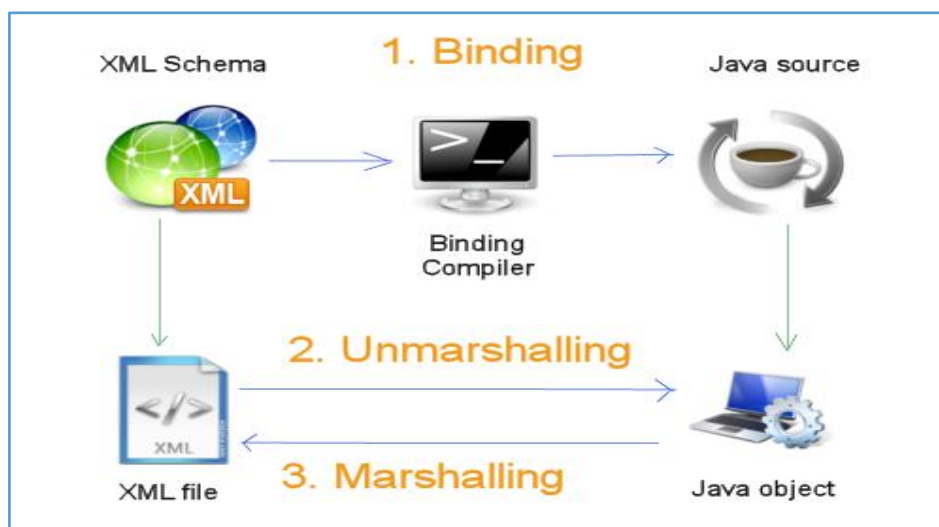
Générer un fichier XML qui correspond à une classe annotée via JAXB

Générer une instance java à partir d'un fichier XML

La mise en œuvre de JAXB 2.0

JAX-B 2.0 est une spécification qui permet de mapper des objets Java dans un document XML et vice versa au moyen d'opérations de sérialisation/désérialisation nommées marshalling/unmarshalling. Il permet aussi de générer des classes Java à partir d'un schéma XML et inversement.

L'API JAX-B est contenue dans le package **javax.xml.bind**



1. Création de la classe

Nous allons générer un document XML à partir de la classe Message

```
@XmlRootElement
public class Message {
    private String from;
    private String to;
    private String text;
    private boolean isNew;
    //getters et setters
}
```

2. Génération d'un fichier XSD à partir de la classe Message

Placez-vous sous le package qui contient la classe Message puis exécutez la commande

« *schemagen Message.java* ».

Un fichier XSD qui porte le nom « **schema1.xsd** » va être généré sous le package où vous avez exécuté la commande.

Analysez le fichier et déduisez les types XSD qui correspondent à chaque attribut de la classe.

❖ Personnalisation des entités

- Ajoutez les annotations JAXB suivante à la classe « Message ».

```
@XmlRootElement
@XmlType(propOrder={"to","text","from"})
@XmlAccessorType(XmlAccessType.PROPERTY)
public class Message {
    private String from;
    private String to;
    private String text;
    private boolean isNew;

    @XmlElement(name="emetteur",required=true)
    public String getFrom() {
        return from;
    }
    public void setFrom(String from) {
        this.from = from;
    }
    @XmlElement(name="destinataire")
    public String getTo() {
        return to;
    }
    public void setTo(String to) {
        this.to = to;
    }
    @XmlAttribute
    public String getText() {
        return text;
    }
    public void setText(String text) {
```

```

        this.text = text;
    }
    @XmlTransient
    public boolean isNew() {
        return isNew;
    }
    public void setNew(boolean isNew) {
        this.isNew = isNew;
    }
}

```

Les annotations utilisées sont :

- **@XmlRootElement** : Associer la classe à l'élément racine du document XML.
- **@XmlType(propOrder)** : Définir l'ordre d'apparition des éléments dans le document XML
- **@XmlElement** : Convertir un attribut en un élément dans le document XML
- **@XmlAttribute** : Convertir un attribut en un attribut dans le document XML
- **@XmlTransient** : Empêcher la sérialisation d'un attribut
- **@XmlAccessorType** : Spécifie si les annotations JAXB seront situées sur les champs (field) ou sur les getters (property)

Analysez l'impact des annotations ajoutées sur le schema produit

3. Génération d'une classe Message à partir du fichier XSD

Placez-vous sous le dossier qui contient le fichier XSD et exécutez la commande « **xjc schema1.xsd** ». Un dossier nommé **generated** va être créé. Il contient la classe Message générée.

Analysez le contenu de la classe et essayez de comprendre le sens des annotations JAXB générées.

4. Génération du fichier xml à partir d'une classe java (Marshalling)

1. Créez la classe suivante qui permet de générer un fichier xml à partir de la classe Message
2. Utiliser la classe **Marshaller** de l'API JAXB pour générer le document XML à partir des objets déjà créés.

```

import java.io.File;
import java.util.ArrayList;
import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Marshaller;
import service.mail.ws.Message;

public class GenererMessageXml {
    public static void main(String[] args) throws JAXBException {
        Message message = new Message();
        message.setFrom("mohamed");
        message.setNew(true);
        message.setText("hello");
        message.setTo("ali");
    }
}

```

```

// création d'un contexte JAXB sur la classe Message
JAXBContext context = JAXBContext.newInstance(Message.class);

// création d'un marshaller à partir de ce contexte
Marshaller marshaller = context.createMarshaller();

// on choisit UTF-8 pour encoder ce fichier
marshaller.setProperty("jaxb.encoding", "UTF-8");
// et l'on demande à JAXB de formater ce fichier de façon
// à pouvoir le lire à l'oeil nu
marshaller.setProperty("jaxb.formatted.output", true);

// écriture finale du document XML dans un fichier message.xml
marshaller.marshal(message, new File("message.xml"));
}

```

Le fichier **XML** généré sera situé sous la **racine du projet**.

5. Génération d'une instance java à partir d'un document xml (Unmarshalling)

Créez et exécutez la méthode main suivante :

```

Public static void main(String[] args) throws JAXBException {
// création d'un contexte JAXB sur la classe Message
JAXBContext context = JAXBContext.newInstance(Message.class) ;
// création d'un unmarshaller
Unmarshaller unmarshaller = context.createUnmarshaller() ;
Message message= (Message)unmarshaller.unmarshal(new File("message.xml")) ;
System.out.println("From = " + message.getFrom()) ;
System.out.println("To = " + message.getTo()) ;
System.out.println("Text = " + message.getText()) ;
}

```