



JAX-B

**Module Services Web
A.U 2021-2022**



Objectifs



➤ **Comprendre le principe de JAX-B**

➤ **Apprendre l'utilisation de JAX-B**



Pla n

- Présentation de JAX-B
- Fonctionnement
- Cas d'usages
- Avantages
- Mapping des types de données
- Utilisation
- Annotations





Présentation de JAX-B

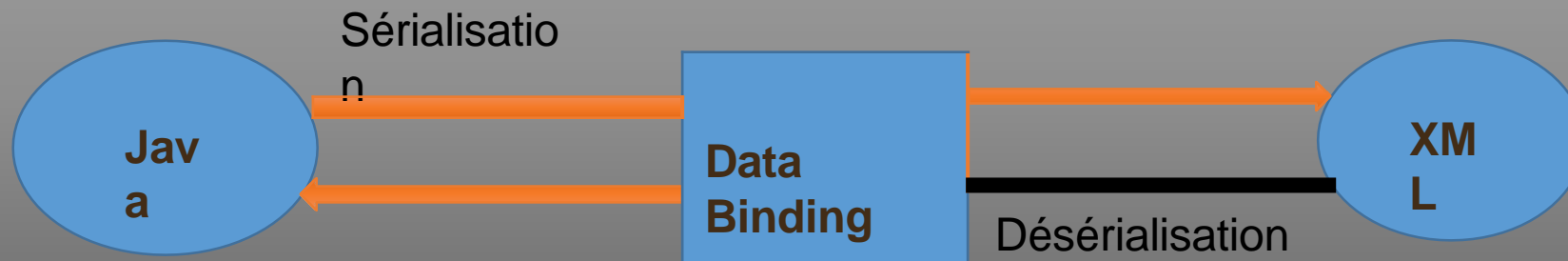


- ❖ Java Architecture for XML Binding
- ❖ JAX-B est un API Java permettant la gestion de données XML.
- ❖ JAX-B 2.0 est incorporée dans Java SE 6
- ❖ JAX-B permet plus particulièrement l'utilisation du "Data Binding"

Présentation de JAXB

Qu'est ce que le « Data Binding » ou association de données ?

Le Data Binding est une technologie permettant d'automatiser la transformation des fichiers XML en objets Java et inversement.

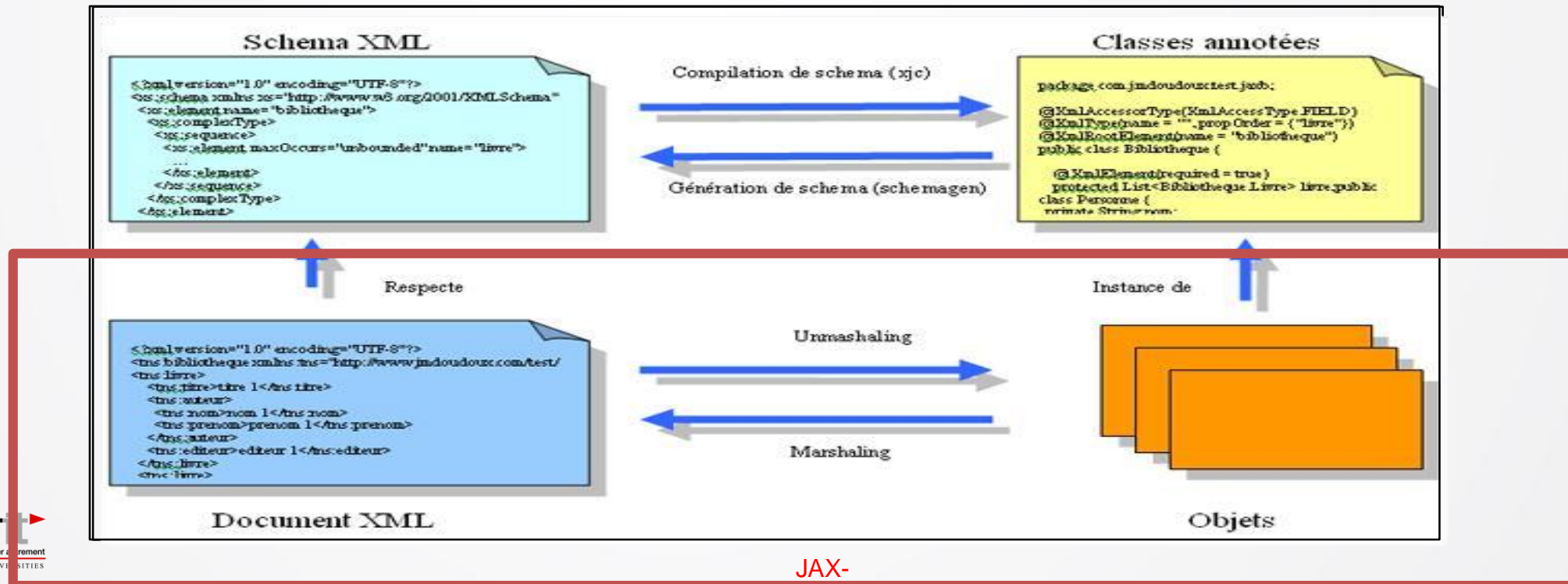


Fonctionnement

JAX-B permet deux types de transformations :

❖ Classes java ↔ Schéma XML

❖ Instances de classes java ↔ XML



Cas d'usages

JAX-B est utilisé par :

❖ JAX-
RS

Les services web RESTful utilisent JAX-B pour la transformation des données échangées en XML



Avantages



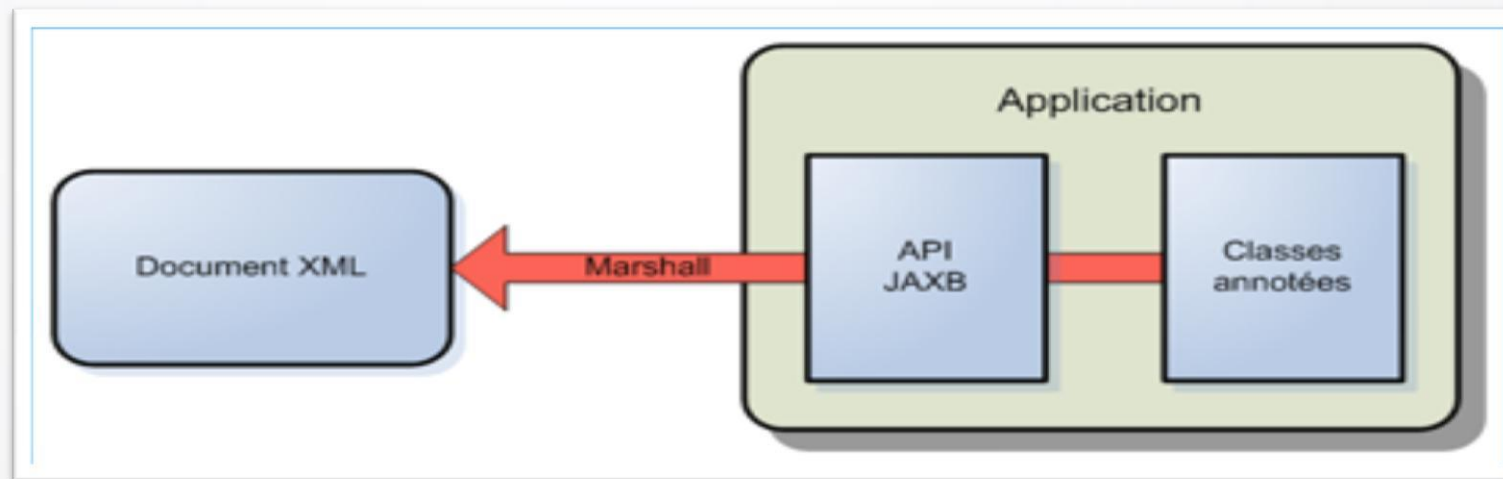
- ❖ Génération de classes automatisée : Gain de temps pour l'utilisateur
- ❖ Meilleure intégrité des données : JAX-B comporte des outils chargés de vérifier l'intégrité des données reçues.

Lorsqu'une erreur intervient, le système pourra lever des exceptions
- ❖ Gestion de la persistance de données stockées sous format XML : La redistribution des données consiste à récupérer le contenu de chaque instance de classe et de les insérer dans les fichiers XML

Utilisatio

Génération d'un document XML à partir d'une instance Java (Sérialisation / Marshalling)

- 1) Ajouter les annotations nécessaires à la classe java
- 2) Utiliser la classe Marshaller de l'API JAX-B pour générer le document XML à partir des objets déjà créés.

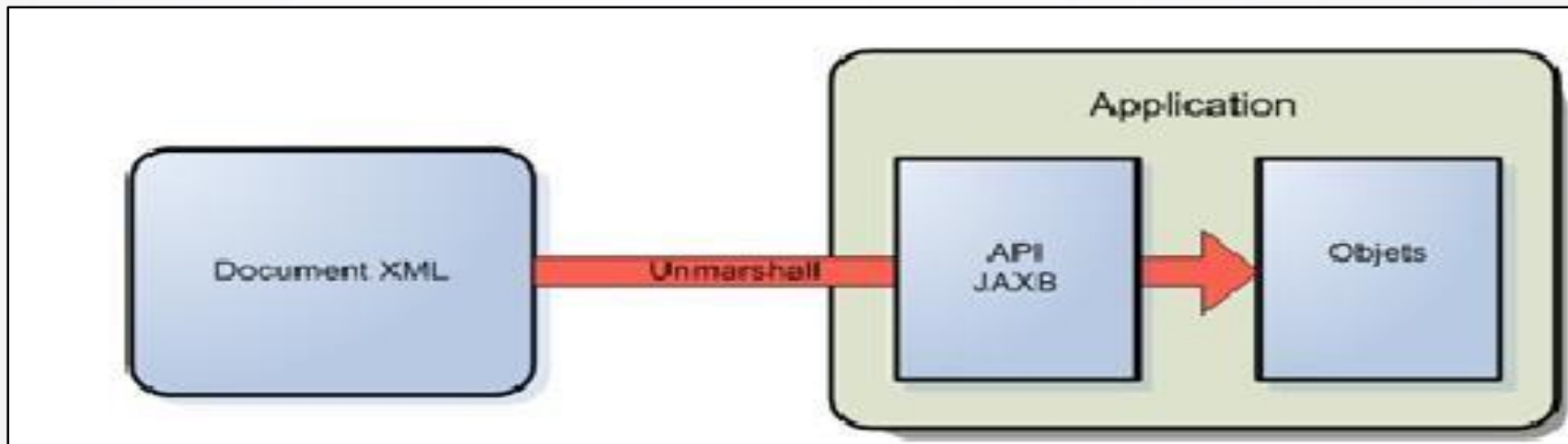


JAX-B

Utilisatio

Mapping d'un document XML à des Objets (Desérialisation / Unmarshalling)

L'API JAX-B propose de transformer un document XML en un ensemble d'objets qui vont encapsuler les données et la hiérarchie du document. Ces objets sont des instances des classes générées à partir du schéma XML.





Annotations

JAX-B



- ❖ Les annotations JAX-B utilisées dans les classes java permettent la génération et la personnalisation des documents XML générés
- ❖ Les annotations JAX-B sont définies dans le package `javax.xml.bind.annotation`.



Annotations

JAX-B

| Annotation | Description |
|-----------------------|---------------------------------------------------------------------------------------|
| XmlRootElement | Spécifier la racine du document XML. |
| XmlElement | Convertir une propriété de la classe en un élément dans le document XML. |
| XmlAttribute | Convertir une propriété de la classe en un attribut dans le document XML. |
| XmlTransient | Retire des éléments pris en compte pour la création des schémas et des documents XML. |



Annotations JAX-B

| Annotation | Description |
|------------------------|---------------------------------------------------------------------------------------------------------------|
| XmlType | Permet de fixer l'ordre dans lequel les champs de cette classe doivent être enregistrés dans le document XML. |
| XmlAccessorType | Contrôler l'ordre des attributs et des propriétés dans la classe. |



Annotations

JAX-B

//Par défaut l'élément XML généré est le nom de la classe « Personne »

@XmlElement (name="maPersonne")

@XmlType(propOrder={"prenom", "nom"})

public class Personne {

private int age;

private String nom;

private String

prenom;

@XmlElement // Par défaut

@XmlElement (name="nom")

public String getNom() { return nom; }

public String getPrenom() { return prenom; }

{

@XmlTransient

public int getAge() { return age; }

}

JAX-B



En Résumé



- ❖ Fournir au développeur un moyen de manipuler un document XML sans connaître XML ou les technologies d'analyse.
- ❖ Manipulations au travers d'objets java.
- ❖ Ces classes sont utilisées pour faire correspondre le document XML dans des instances de ces classes et vice et versa : ces opérations se nomment respectivement unmarshalling et marshalling.

En

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <formes>
3   <cercle>
4     <centre x="100" y="73"/>
5     <rayon>25</rayon>
6   </cercle>
7   <carré>
8     <centre x="147" y="111"/>
9     <côté>100</côté>
10  </carré>
11  <cercle>
12    <centre x="196" y="92"/>
13    <rayon>50</rayon>
14  </cercle>
15 </formes>
```

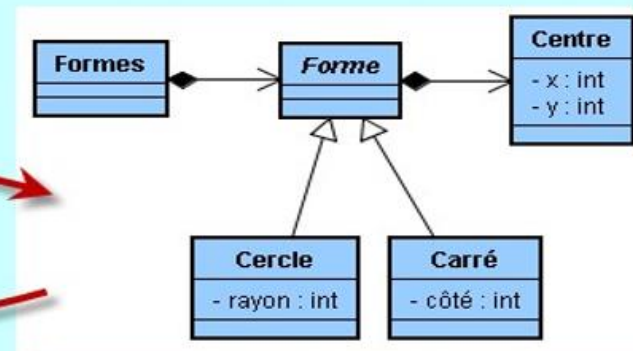
Document XML

JAX-B

Classes annotées

```
// Définition de l'élément formes et de ses sous-éléments
@XmlRootElement
public class Formes {
    @XmlElement({
        @XmlElement(name = "carré", type = Carré.class),
        @XmlElement(name = "cercle", type = Cercle.class)
    })
    private ArrayList<Forme> formes = new ArrayList<Forme>();
    public ArrayList<Forme> getFormes() { return formes; }
    public void ajoutForme(Forme forme) { formes.add(forme); }
    public void supprimerFormes() { formes.clear(); }
}
```

Instances de



Objets issus des classes