

Introduction à JAXB, JAX-WS et WSDL

URI (Uniform Resource Identifier) : Un URI doit permettre d'identifier une ressource de manière permanente, même si la ressource est déplacée ou supprimée. Cette ressource peut être de deux formes soit un emplacement soit un nom

URL (Uniform Resource Locator) : Une URL est une URI qui permet de localiser une ressource identifiée ainsi que le moyen d'accéder à cette ressource. Les URLs sont soumis à des normes syntaxiques décrites dans la RFC 3986. Elle peut localiser plusieurs formes de données : html, image,

URN (Uniform Resource Name) : l'URN donne une syntaxe de chaîne de caractères utilisable pour identifier une ressource (un document, une image, un enregistrement sonore, etc.) globalement, durant toute son existence, indépendamment de sa localisation ou de son accessibilité par Internet.
syntaxe : urn:NID:NSS ("NID = Namespace Identifier" et "NSS = Namespace Specific String")

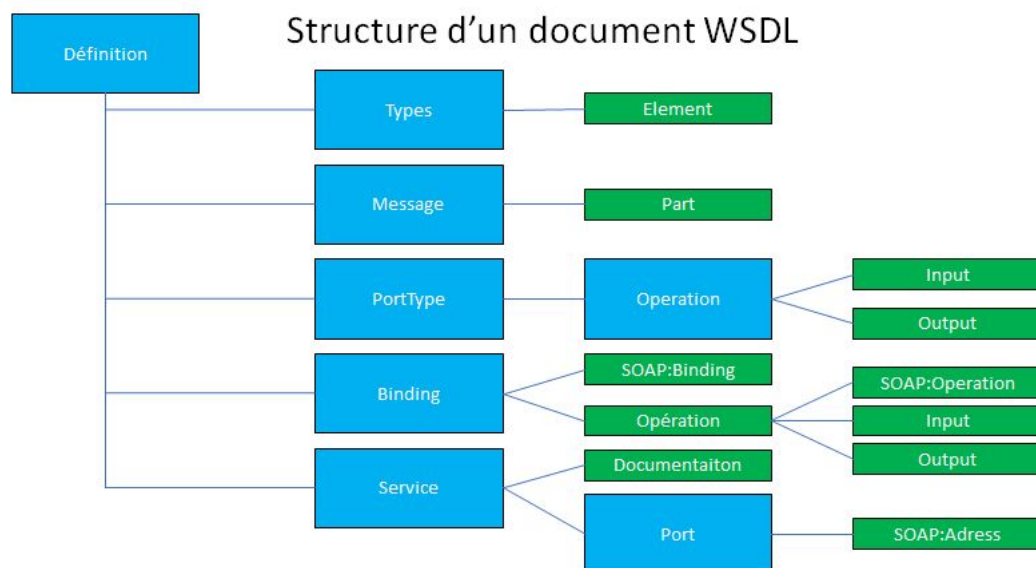
WSDL (Web Services Description Language) :

Un document WSDL décrit le type d'un service web (méthodes, types des paramètres,...) mais aussi les aspect technique d'implémentation d'un service web (protocoles utilisé, adresse du service,...). La description WSDL est un document XML qui commence par la balise définition.

Exemple :

```
<definitions xmlns="http://www.w3.org/ns/wsd1"
  xmlns:tns="http://www.tmsws.com/wsd120sample"
  xmlns:whhttp="http://schemas.xmlsoap.org/wsd1/http/"
  xmlns:wsoap="http://schemas.xmlsoap.org/wsd1/soap/"
  targetNamespace="http://www.tmsws.com/wsd120sample">
...
</definitions>
```

Schéma de la structure d'un document WSDL :



- **Types** : fournit types de données utilisés par les messages.
- **Messages** : cette balise décrit la structure d'un message échangé (nom, type)
- **PortTypes** : cette balise décrit un ensemble d'opérations. Chaque opération à zéro ou un message en entrée, zéro ou plusieurs messages de sortie ou d'erreurs.
- **Binding** : spécifie une liaison entre un <portType> et un protocole concret (SOAP, HTTP...).
- **Service** : cette balise décrit un service web comme un ensemble de port.
- **Port** : représente un point d'accès de services défini par une adresse réseau et une liaison.
- **Opération** : cette balise décrit une opération réalisée par le service web. Elle permet d'envoyer et de recevoir des messages.

Annotations JAXB :

- **@XmlRootElement** : associe la classe annotée avec un nœud racine d'un document XML.
 - **@XmlRootElement(name="nom")** permet de changer le nom de l'élément principal
- **@XmlAttribute** : permet d'écrire le champ de classe dans un attribut XML plutôt que dans un sous-élément de l'élément XML parent.
 - **@XmlAttribute(name="label")** permet de changer le nom de l'attribut.
- **@XmlElement** : associe un champ de classe à un nœud d'un document XML.
 - **@XmlElement(name="label")** permet de changer le nom de l'élément.
 - C'est l'annotation par défaut pour les champs d'une classe
- **@XmlTransient** : fait en sorte de retirer le champ de classe des éléments pris en compte pour la création des schémas et des documents XML.
- **@XmlType(propOrder={"champ3", "champ2", "champ1"})** : permet de définir l'ordre dans lequel les champs de cette classe doivent être enregistrés dans le document XML (uniquement les champs avec l'annotation **@XmlElement**). Il faut que les nom de champs soient ceux des champs et non ceux inscrits dans l'annotation **@XmlElement(name="label")**.
- **@XmlElementWrapper** et **@XmlElements** : Si le champ est de type List, on peut spécifier que chaque élément de la liste, du fait de sa classe, est associée à un nœud différent.
 - **@XmlElementWrapper(name="boiteOutils")**
@XmlElements({
@XmlElement(name="capitaine", type=Marteau.class),
@XmlElement(name="cuisinier", type=Tournevis.class)
})
Ici il des instances des classes Marteau et Tournevis sont contenues dans une liste dont nous nommons l'élément XML lié "boiteOutils".

Annotations JAX-WS :

- **@WebService** : marque une classe java comme implémentant un service web.
- **@WebFault** : mappe les exceptions java en wsdl:fault
- **@WebServiceRef** : définit une référence à un service Web appelé par le client.
- **@WebServiceProvider** : indique qu'une classe satisfait les exigences d'une classe d'implémentation de fournisseur JAX-WS. Incompatible avec **@WebService**.
- **@WebParam** : personnalise le mappage d'un paramètre individuel vers un élément XML.
- **@WebMethod** : indique une méthode correspondant à une opération de service Web.

```
targetNamespace="http://www.localhost:8989/" name="WebServiceSoapService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://www.localhost:8989/" schemaLocation="http://localhost:8080/?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="multi">
    <part name="parameters" element="tns:multi"/>
  </message>
  <message name="multiResponse">
    <part name="parameters" element="tns:multiResponse"/>
  </message>
  <portType name="WebServiceSoap">
    <operation name="multi">
      <input wsam:Action="http://www.localhost:8989/WebServiceSoap/multiRequest" message="tns:multi"/>
      <output wsam:Action="http://www.localhost:8989/WebServiceSoap/multiResponse" message="tns:multiResponse"/>
    </operation>
  </portType>
  <binding name="WebServiceSoapPortBinding" type="tns:WebServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="multi">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="WebServiceSoapService">
    <port name="WebServiceSoapPort" binding="tns:WebServiceSoapPortBinding">
      <soap:address location="http://localhost:8080/">
    </port>
  </service>
</definitions>
```

<http://localhost:8080/?wsdl>

```
targetNamespace="http://www.localhost:8989/">
  <xs:element name="add" type="tns:add"/>
  <xs:element name="addResponse" type="tns:addResponse"/>
  <xs:element name="multi" type="tns:multi"/>
  <xs:element name="multiResponse" type="tns:multiResponse"/>
  <xs:complexType name="add">
    <xs:sequence>
      <xs:element name="premierTerme" type="xs:double"/>
      <xs:element name="secondTerme" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="addResponse">
    <xs:sequence>
      <xs:element name="résultat" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="multi">
    <xs:sequence>
      <xs:element name="premierFacteur" type="xs:double"/>
      <xs:element name="secondFacteur" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="multiResponse">
    <xs:sequence>
      <xs:element name="résultat" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

<http://localhost:8080/?xsd=1>