

Série TD 3 (2 séances)

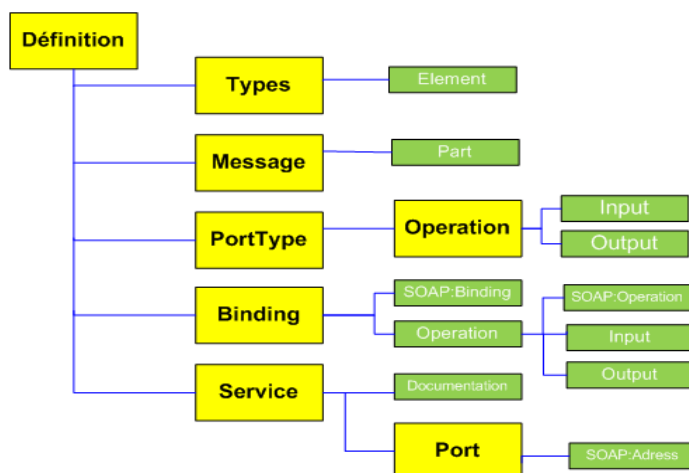
WSDL – Décrire, configurer, et personnaliser un Service Web

Structure d'un fichier WSDL

Le WSDL -Web Service Description Language- est une grammaire XML permettant de décrire des services web afin d'indiquer à un consommateur la manière de communiquer et d'utiliser ces derniers. Il fournit une description indépendante du langage et de la plateforme.

A partir d'un document WSDL, il est possible : **a)** de générer un client pour appeler un service web ; **b)** de générer le code pour implémenter un service web.

Un document WSDL se compose d'un ensemble d'éléments décrivant les types de données utilisés par le service web, les messages que le service peut recevoir, ainsi que les liaisons SOAP associées à chaque message.¹



Structure d'un document WSDL

```
<wsdl:definitions>
  <wsdl:types />
  <wsdl:message />
  <wsdl:portType>
    <wsdl:operation>
      <wsdl:input />
      <wsdl:output />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding />
  <wsdl:service />
</wsdl:definitions>
```

Concepts d'un document WSDL

Types : représente la description des types de données des messages échangés.

Message : décrit les données échangées par le service web – Les paramètres des opérations.

PortType : - interface au sens Java – décrit l'ensemble des opérations du service web.

Operation : - méthode au sens Java – c'est la description d'une action exposée dans le port.

Binding : définit pour un type de port le protocole utilisé (http, soap, etc.) pour transmettre les messages et les formats de données.

Service : définit un ou plusieurs ports.

Port : définit où est localisé le service web (l'adresse url) et le binding à utiliser.

Les trois derniers concepts représentent la *partie concrète* d'un document WSDL, tandis que les autres représentent sa *partie abstraite*.

¹ Source figure – Structure document WSDL : <https://openclassrooms.com/courses/les-services-web>

Annotations et personnalisation du WSDL

JAX-WS repose sur l'utilisation massive d'annotations pour configurer un service web et customiser son document WSDL.

Les principales annotations sont les suivantes :

- **@WebService** : annote une interface/classe Java pour décrire/implementer un service web (annotation obligatoire).
- **@WebMethod** : Paramétrer une opération.
- **@WebParam** : Paramétrer un message.
- **@WebResult** : Paramétrer un message de sortie.
- **@OneWay** : input seulement.

Chaque annotation peut être configurée grâce à ses attributs ²: exemples :

@WebService	<ul style="list-style-type: none">○ String <i>name</i> : pour changer le nom du service web (portType).○ String <i>endpointInterface</i> : pour changer le nom de l'interface décrivant le service web.○ String <i>portName</i> : pour changer le nom du port.○ String <i>serviceName</i> : pour changer le nom de service du service web.
@WebMethod	<ul style="list-style-type: none">○ boolean <i>exclude</i> : précise que la méthode ne doit pas être exposée comme une opération.○ String <i>operationName</i> : précise le nom de l'attribut name défini dans l'élément operation du document WSDL.
@WebParam	String <i>name</i> : nom du paramètre (dans wsdl:types).

Exercice 1

Ecrire un service web Convertisseur fournissant deux opérations pour convertir un montant donné en paramètre :

- Une opération **getDinarFromEuro** : pour convertir de l'euro au dinar.
- Une opération **getEuroFromDinar** : pour convertir du dinar à l'euro.

1. Ecrire les classes de ce service web, à savoir, Convertisseur (interface), ConvertisseurImpl, et ConvertisseurPublisher.
2. Ecrire une classe cliente qui consomme ce service web en appelant l'opération 1 et 2.
3. Changer depuis l'interface le nom du service web en ConvertisseurDinarEuro.
4. Changer depuis l'interface le nom de l'opération 2 en DinarVersEuro.
5. Changer depuis l'interface le nom du paramètre de l'opération 1 en montantEuro et le nom de son résultat retourné en montantDinar.

² <http://mbaron.developpez.com/cours/soa/wsdl/>

Exercice 2

Ecrire un service web ShopCatalogWebService fournissant des opérations pour la gestion de produits d'un magasin. Une opération pour ajouter un produit, une autre pour récupérer la liste complète des produits et enfin une dernière opération pour récupérer un produit par un identifiant. Un produit est décrit par un identifiant (String), une catégorie (String), et un prix.

- 1 Ecrire les classes de ce service web, à savoir, Product, ShopCatalog (interface), ShopCatalogImpl, et ShopCatalogPublisher.
- 2 Changer depuis l'interface le nom du service web en CatalogMagasin.
- 3 Changer depuis l'interface le nom de l'opération 2 en *tousLesProduits*.
- 4 Changer depuis l'interface le nom du paramètre de l'opération 3 en *produitID* et le nom du résultat retourné en *produitRetourné*.
- 5 Pour tester ce service web, créer une classe cliente qui le consomme en appelant l'opération 2.