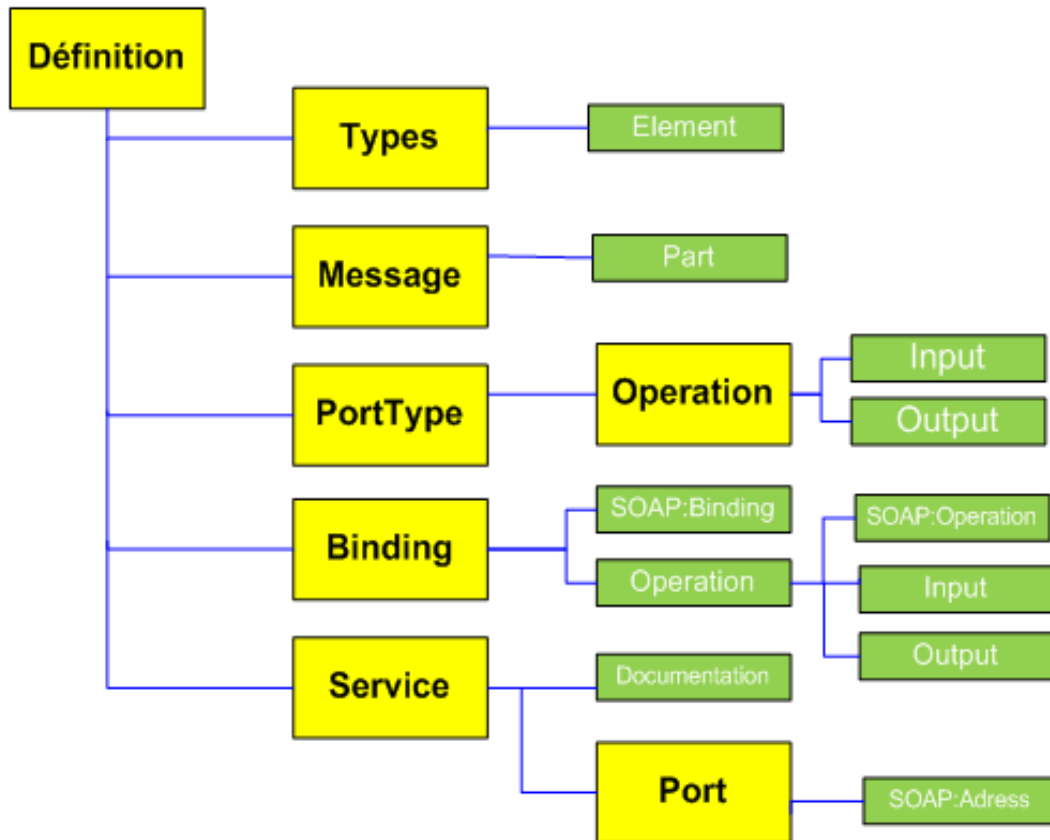


Les Services Web



WSDL – Décrire, configurer, et personnaliser un Service Web

Contrat Service Web - WSDL



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>

Contrat Service Web - WSDL

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/implémenter un service web (annotation obligatoire).
- **@WebMethod** : Paramétrer une opération.
- **@WebParam** : Paramétrer un message paramètre.
- **@WebResult** : Paramétrer un message de sortie (return).
- **@OneWay** : input seulement.

Contrat Service Web - WSDL

1. **@WebService**: *name*, *serviceName*, *portName*

```
@WebService(name="Hello")
```

```
public interface HelloWorld {
```

```
    @WebMethod
```

```
    public String simpleHello();
```

```
    @WebMethod
```

```
    public String makeHello(String name);
```

```
}
```

```
<wsdl:portType>
```

```
<wsdl:operation>
```

```
<wsdl:input />
```

```
<wsdl:output />
```

```
</wsdl:operation>
```

```
</wsdl:portType>
```

Contrat Service Web - WSDL

1. **@WebService**: *name*, ***serviceName***, ***portName***

```
@WebService(endpointInterface = "org.soa.ws.tp.HelloWorld",  
             serviceName="HelloService",  
             portName="HelloPort")
```

```
public class HelloWorldImpl implements HelloWorld{
```

```
    @Override
```

```
    @WebMethod
```

```
    public String simpleHello() {  
        return "Hello World";  
    }
```

```
}
```

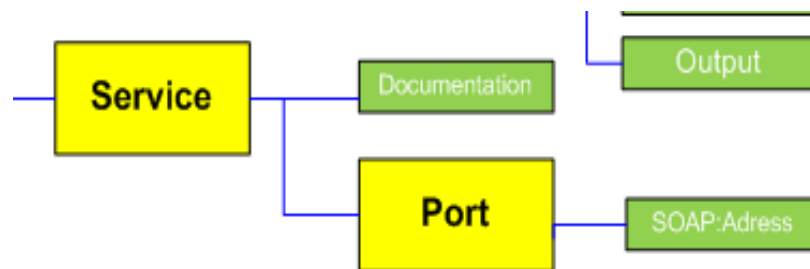
```
    @Override
```

```
    @WebMethod
```

```
    public String makeHello(String name) {  
        return "Hello World, " + name;  
    }
```

```
}
```

```
}
```



Contrat Service Web - WSDL

1. **@WebMethod**: **operationName**, *exclude*

```
@WebService(name="Hello")  
public interface HelloWorld {
```

```
    @WebMethod(operationName="operationSimple")  
    public String simpleHello();
```

```
    @WebMethod(operationName="operationMake")  
    public String makeHello(String name);
```

```
}
```

```
<wsdl:portType>  
    <wsdl:operation>  
        <wsdl:input />  
        <wsdl:output />  
    </wsdl:operation>  
</wsdl:portType>
```

Contrat Service Web - WSDL

1. **@WebMethod**: *operationName*, **exclude**

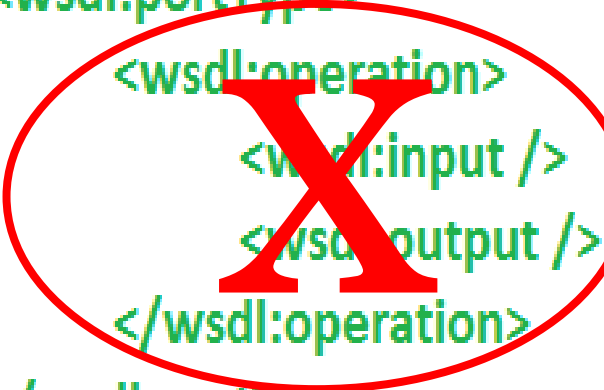
```
@WebService(name="Hello")  
public interface HelloWorld {
```

```
    @WebMethod(operationName="operationSimple")  
    public String simpleHello();
```

```
    @WebMethod(exclude=true)  
    public String makeHello(String name);
```

```
}
```

```
<wsdl:portType>  
  <wsdl:operation>  
    <wsdl:input />  
    <wsdl:output />  
  </wsdl:operation>  
</wsdl:portType>
```



Contrat Service Web - WSDL

1. @OneWay:

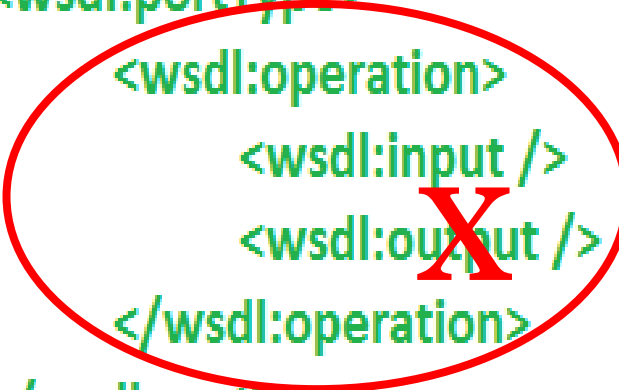
```
@WebService(name="Hello")
public interface HelloWorld {

    @WebMethod
    public String simpleHello();

    @WebMethod
    @OneWay
    public String makeHello(String name);

}
```

```
<wsdl:portType>
  <wsdl:operation>
    <wsdl:input />
    <wsdl:output />
  </wsdl:operation>
</wsdl:portType>
```



Contrat Service Web - WSDL

1. @WebParam: *name*

<wsdl:types />
<wsdl:message />

```
@WebService(name="Hello")  
public interface HelloWorld {
```

```
    @WebMethod(operationName="operationSimple")  
    public String simpleHello();
```

```
    @WebMethod  
    public String makeHello(@WebParam(name="nom") String name);  
}
```

Contrat Service Web - WSDL

1. @WebResult: *name*

<wsdl:types />
<wsdl:message />

```
@WebService(name="Hello")  
public interface HelloWorld {
```

```
    @WebMethod(operationName="operationSimple")  
    public String simpleHello();
```

```
    @WebMethod
```

```
    @WebResult(name="MonResultat")
```

```
    public void makeHello(@WebParam(name="nom") String name);
```

```
}
```

Contrat Service Web - WSDL

Exercice 1 – Série TD 3

Ecrire un service web Convertisseur fournissant deux opérations :

- 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.

<https://github.com/GitTeaching/ConvertisseurWS/tree/master/src/org/soa/ws/tp3>

<https://github.com/GitTeaching/ConvertisseurWSClient/blob/master/src/ConvertisseurWSClient.java>

Contrat Service Web - WSDL

Exercice 1 – Série TD 3

Changer depuis l'interface le nom du service web en ConvertisseurDinarEuro.

Changer depuis l'interface le nom de l'opération 2 en dinarVersEuro.

Changer depuis l'interface le nom du paramètre de l'opération 1 en montantEuro et le nom de son résultat retourné en montantDinar.

```
@WebService(name="ConvertisseurDinarEuro")
```

```
public interface Convertisseur {
```

```
    @WebMethod
```

```
    @WebResult(name="montantDinar")
```

```
    public double getDinarFromEuro(@WebParam(name="montantEuro")  
                                   double euro);
```

```
    @WebMethod(operationName="dinarVersEuro")
```

```
    public double getEuroFromDinar(double dinar);
```

```
}
```

Contrat Service Web - WSDL

Exercice 1 – Série TD 3

Client - Changement

```
public class ConvertisseurWSClient {  
  
    public static void main(String[] args) {  
        ConvertisseurImplService service = new ConvertisseurImplService();  
  
        ConvertisseurDinarEuro convertisseur =  
            service.getConvertisseurImplPort();  
  
        System.out.println(convertisseur.dinarVersEuro(100));  
    }  
}
```