

Services Web

Cours 3

Isis TRUCK
Université Paris 8

Références

- <http://www-igm.univ-mlv.fr/~dr/XPOSE2004/woollams/index.html>
- <https://benoitpiette.com/labo/introduction-aux-web-services.html>
- <http://www.christian-faure.net/2012/10/08/services-web-ou-api/>
- <http://www.gchagnon.fr/cours/xml/servicesweb.html>
- <https://openclassrooms.com/courses/les-requetes-http>
- <https://doc.ubuntu-fr.org/wget>
- <https://fr.wikipedia.org/>
- https://www.w3schools.com/xml/xml_dtd_intro.asp
- <https://www.irif.fr/~carton/Enseignement/XML/Cours/Namespace/index.html>
- Et beaucoup d'autres...

Plan

- Cours 1 : Préliminaires : couches réseau
 - Modèle théorique : OSI
 - Modèle pratique : TCP/IP
 - HTTP
- Cours 2 : Services Web ou API ?
 - Service Web (Génèse, ..., XML, DTD, XSD...)
 - API, API publique
 - Contrat
- **Cours 3 : REST, SOAP, WSDL, UDDI**

Service Web, comme une pile de protocoles réseau

- ensemble de protocoles permettant à des applications de communiquer entre elles, indépendamment de la structure et du langage employés

Découverte de services	ex. UDDI
Description de services	ex. WSDL
Communication	ex. SOAP
Transport	ex. HTTP

Service Web, comme une pile de protocoles réseau

- Il y a deux types d'architecture derrière les services Web :
 - REST (*REpresentational State Transfer*)
 - SOA (*Service Oriented Architectures*)
- Le modèle **REST** approfondit les notions de ressource et d'état ; lire la page *Wikipedia* sur REST et la notion de serveur *stateless* (chaque requête est vue comme une transaction indépendante)
 - REST repose entièrement et uniquement sur HTTP
 - REST est (aussi) beaucoup utilisé, par ex. les services Web d'Amazon sont en REST

Service Web, comme une pile de protocoles réseau

- Les architectures orientées service (SOA) mettent en avant un **couplage** dit **lâche** (ou **faible**) entre client et fournisseur de service
- Couplage **faible** car il s'agit de la communication entre 2 composants physiquement séparés
- Les SOA découplent ainsi une application répartie en deux couches :
 - Une couche d'**offre** de services spécifiés par leurs interfaces et éventuellement des informations de nature sémantique dans un annuaire de services (UDDI)
 - Une couche de **réalisation physique** des services sur des machines connectées à Internet et accessibles via des protocoles standards

Service Web, comme une pile de protocoles réseau

- Pour établir ou **consommer** un service Web, plusieurs méthodes de communication sont utilisées :
 - Au départ, RPC (*Remote Procedure Call*)
 - Puis XML-RPC
 - Puis JSON-RPC
 - et SOAP (*Simple Object Access Protocol*)
- Ces protocoles permettent à des processus issus d'environnements hétérogènes, **l'accès à des fonctions** (procédures) **distantes** via un protocole de communication (**HTTP**, SMTP...)
 - avec SOAP, le format d'échange est du XML
 - avec JSON, le format d'échange est du Javascript (JSON = *JavaScript Object Notation*)

SOAP

- SOAP utilise le plus souvent HTTP comme protocole de communication, mais il peut aussi utiliser d'autres protocoles comme SMTP, TCP, UDP...
- SOAP est un des protocoles les plus répandus pour réaliser des services Web
- Il est géré en natif en **PHP** depuis la version 5.0.1
- Exemple d'une procédure SOAP :
 - Une application envoie une requête SOAP à un serveur qui offre des services Web, comme une base de données des prix de l'immobilier, avec un certain nombre de paramètres
 - Le serveur renvoie une réponse SOAP (un document formaté XML avec les données de réponse), par ex. des prix, des lieux et des caractéristiques des biens immobiliers.
 - Etant donné que les données générées arrivent dans un format *parsable* automatiquement, l'application cliente peut intégrer la réponse directement (dans son code).

SOAP

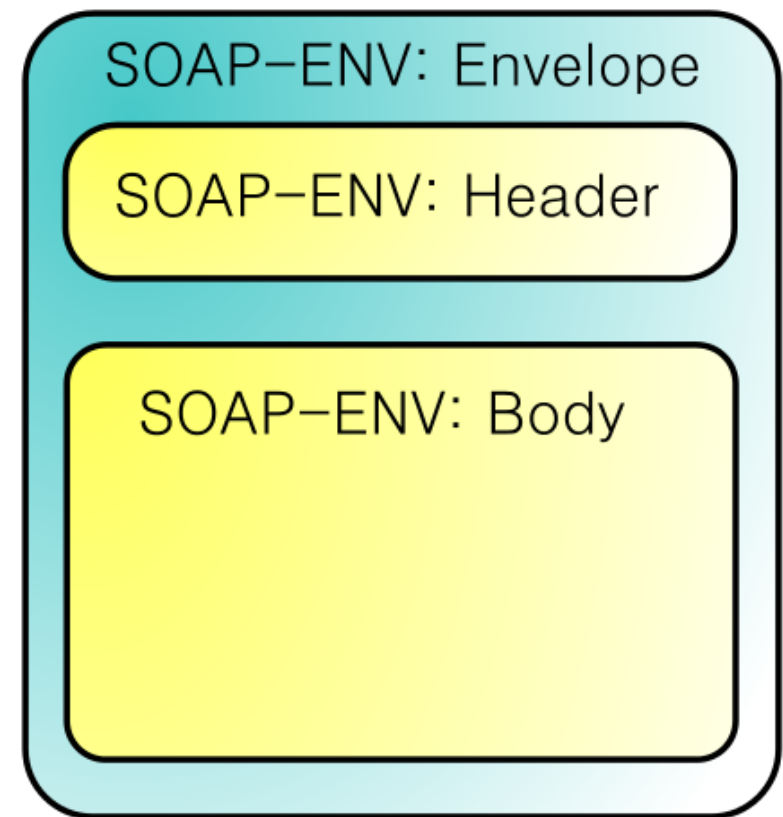
- La spécification SOAP consiste en 3 composants conceptuels :
 - Concepts de **protocole**,
 - Concepts d'**encapsulation** et
 - Concepts de **réseau**
- Les concepts d'encapsulation des données définissent un **message**, une **enveloppe**, un **bloc d'en-tête**, une **en-tête**, un **corps** et un élément **erreur**
- Un message SOAP est donc un document XML contenant :
 - *Envelope* : Identifie le document XML document comme un message SOAP. Obligatoire.
 - *Header* : contient l'info d'en-tête. Optionnel.
 - *Body* : contient l'appel et l'information de réponse. Obligatoire
 - *Fault* : Fournit des informations à propos des erreurs qui se sont produites lors du traitement du message. Optionnel

SOAP

- Ex. de message de requête SOAP :

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml;
charset=utf-8
Content-Length: 299
SOAPAction: "http://www.w3.org/2003/05/soap-
envelope"
```

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://
www.w3.org/2003/05/soap-envelope"
xmlns:m="http://www.example.org/stock/
Manikandan">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <m:GetStockPrice>
      <m:StockName>GOOGLE</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```



Source: <https://en.wikipedia.org/wiki/SOAP>

Voir un exple complet ici : http://w3schools.sinsixx.com/soap/soap_example.asp.htm

WSDL

- SOAP est donc le protocole d'échange entre 2 objets distants
 - 1^{er} objet : serveur contenant le service Web
 - 2^e objet : client voulant consommer le service Web
- Comment savoir ce que l'on peut consommer, et comment ?
 - => C'est l'objet de la description WSDL
- Le WSDL (*Web Services Description Language*) doit être fourni par le serveur fournisseur du service Web (le serveur)
- On peut théoriquement s'en passer, mais la fourniture du service Web risque d'être difficile, si l'on ne sait pas ce que le service peut apporter...
- Le fichier WSDL est en XML et il permet de **décrire l'utilisation complète** du service, comme un **mode d'emploi**
- il permet aux programmes de découvrir automatiquement un service, voire de gérer les modalités d'échanges SOAP (port d'échange, protocole, encodage, etc.)
- WSDL peut être comparé à la DTD d'un fichier XML (des outils permettent la réalisation assistée d'un WSDL, comme Spring-WS)

WSDL

- Une description WSDL est donc un document XML qui commence par la balise <definitions> et qui contient les balises suivantes :
 - <binding> : définit le protocole à utiliser pour invoquer le service web
 - <port> : spécifie l'emplacement effectif du service
 - <service> : décrit un ensemble de points finaux du réseau

WSDL

- WSDL commence par permettre à un utilisateur de décrire de nouveaux types de données dans une syntaxe XML Schema (cf. les trois lignes suivantes dans la description générale de la structure d'un document WSDL) :

```
<wsdl:types>  
  <xsd:schema> ... </xsd:schema>  
</wsdl:types>
```

- WSDL décrit ensuite la structure d'un ensemble de messages (messages de requêtes et de réponse)

```
<wsdl:message name="nmtoken">  
  <part name="nmtoken" element="qname"? type="qname"?/  
>  
</wsdl:message>
```

- Puis, WSDL définit un élément « type de port » comme un ensemble d'opérations. Une opération est définie par un message d'appel (input) et un message résultat (output). Le standard prévoit aussi la description d'une structure de données associée aux erreurs.

WSDL

```
<wsdl:portType name="nmtoken">
  <wsdl:operation name="nmtoken">
    <wsdl:input name="nmtoken"? message="qname">
    </wsdl:input>
    <wsdl:output name="nmtoken"? message="qname">?
    </wsdl:output>
    <wsdl:fault name="nmtoken" message="qname">
    </wsdl:fault>
  </wsdl:operation>
</wsdl:portType>
```

- L'élément binding (liaison) définit ensuite concrètement comment accéder à un service en indiquant son type de port, le protocole utilisé pour y accéder (c'est le plus souvent SOAP en mode HTTP) et la localisation sur le Web de ce service (URI de l'endroit où se trouve le service web)

WSDL

```
<wsdl:binding name="nmtoken" type="qname">
  <wsdl:operation name="nmtoken">
    <wsdl:input>
    </wsdl:input>
    <wsdl:output>
    </wsdl:output>
    <wsdl:fault name="nmtoken">
    </wsdl:fault>
  </wsdl:operation>
</wsdl:binding>
```

- Finalement le service rendu sur le Web par une entreprise peut être composé de plusieurs ports c'est à dire de plusieurs applications accessibles sur le web (correspondant à des *bindings*) et donc localisées éventuellement sur des serveurs différents

```
<wsdl:service name="nmtoken">
  <wsdl:port name="nmtoken" binding="qname">
  </wsdl:port>
</wsdl:service>
```

WSDL

- Exemple complet sur

https://www.tutorialspoint.com/wsdl/wsdl_example.htm

```
<definitions name = "HelloService"
  targetNamespace = "http://www.examples.com/wsdl/HelloService.wsdl"
  xmlns = "http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns = "http://www.examples.com/wsdl/HelloService.wsdl"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema">

  <message name = "SayHelloRequest">
    <part name = "firstName" type = "xsd:string"/>
  </message>

  <message name = "SayHelloResponse">
    <part name = "greeting" type = "xsd:string"/>
  </message>
  ...
```


WSDL

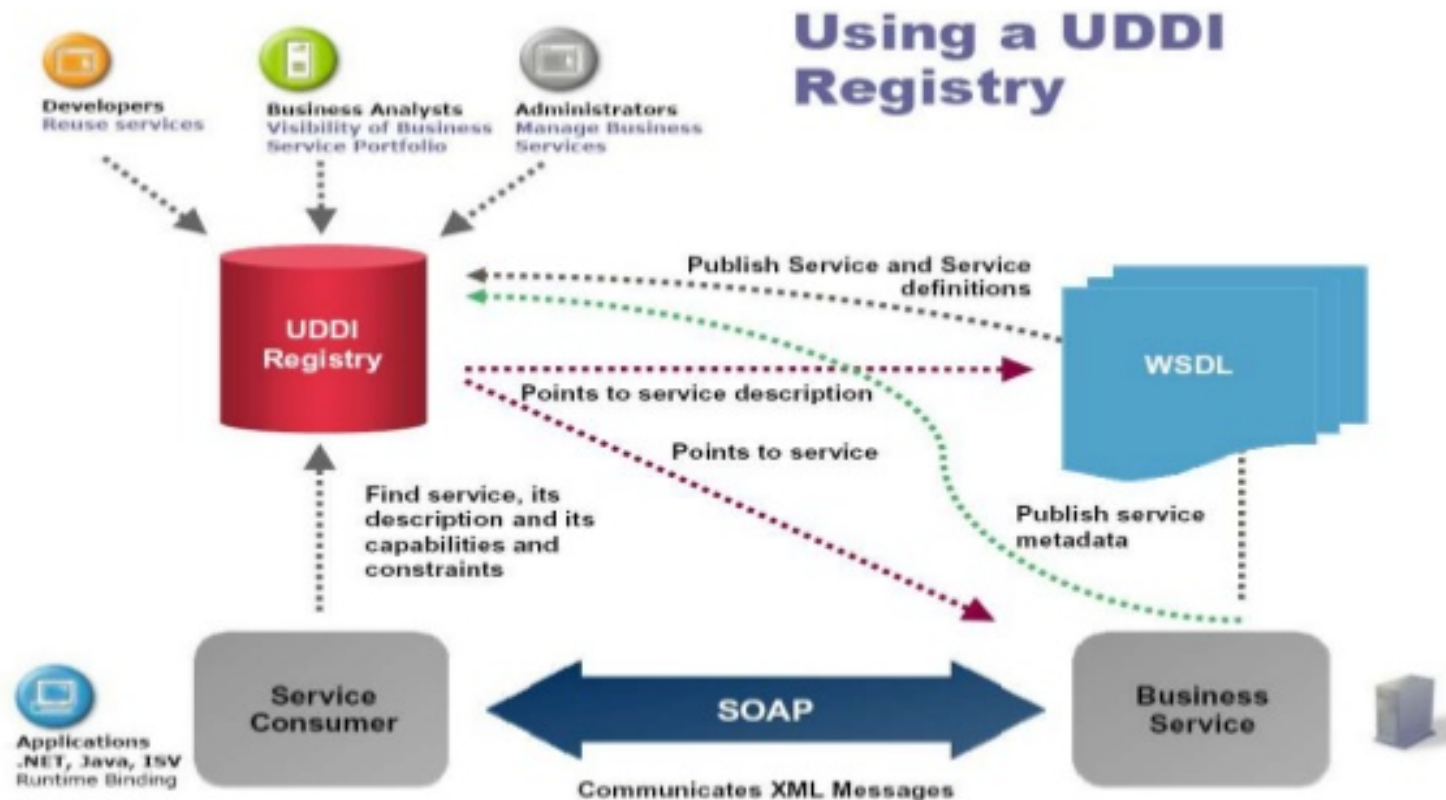
- PHP, à partir de la version 5, interprète de façon transparente les fichiers WSDL lors de la déclaration d'un client à un service Web, et gère lui-même le formatage SOAP des messages envoyés au serveur : formatage XML, respect des règles du WSDL, etc.
- *Voir TP n°6*

UDDI

- ***Universal Description, Discovery, and Integration***
- Service d'annuaire pour trouver un service Web
- L'annuaire détient les informations du service Web
- Le fournisseur du service Web définit son service (via WSDL, par exemple) et ensuite le **publie** dans l'annuaire pour qu'il soit accessible aux clients

UDDI

Overview of UDDI Registry



UDDI

How UDDI Works

- A technical specification for publishing and finding businesses and Web services.

