



Langages de l'internet

Introduction aux Web services et à l'Architecture REST

Bruno Mascret

CPE Lyon

Techniques et langages de l'Internet

Plan

- 1 Les Services Web
 - Principes
 - Sémantique des web service
 - Quelques ressources

Introduction

Auteurs

Cours : **Bruno Mascret**

TPs : **Bruno Mascret**

Intervenants : **Bruno Mascret, Jacques Saraydaryan, Dino Cosmas**

Plan

- 1 Les Services Web
 - Principes
 - Sémantique des web service
 - Quelques ressources

Les Services Web

Principes

- ≡ Encapsuler des services réseaux sur le web
 - utilisation du port 80 pour contourner les pare-feux ;
 - publier facilement des algorithmes, en interne et/ou à l'externe.
- ≡ délégation de compétence : de « je fais » à « je connais quelqu'un qui fait mieux que moi » ;
- ≡ permettre le dialogue entre machines en limitant les interactions humaines.

Les Services Web

Deux approches

Il existe deux manières principales d'appréhender un service web :

1. approche client–serveur ;
2. approche ressource/service.

La 1ère vient d'une continuité historique avec les infrastructure clients–serveurs (héritée de CORBA, RMI, CGI, servlets, etc.).

On parlera de **services** et de **client–serveur**.

La seconde vient directement du monde du WEB, on parlera **d'agents** et de **ressources**.

Ces deux approches décrivent la même chose dans des paradigmes différents.

Toutefois, la mise en œuvre des solutions sera très souvent colorée par l'approche qui a pré–dominée dans la conception.

Les Services Web

Notion de web sémantique

Les services web contribuent à la notion de Web sémantique.

- ≡ idée (utopique ?) qu'un sens peut être donné à toute ressource ;
- ≡ qu'une ressource peut-être auto-explicative ;
- ≡ qu'un usage peut être décrit ;
- ≡ qu'un agent ou une machine peut interpréter ce sens et cet usage.

Concrètement, les web services devront être décrit afin de pouvoir être utilisés comme leur concepteur l'a imaginé.

Sémantique des web service

Description d'un web service

Décrire un web service c'est donner :

- ≡ Une description des fonctions « **exposées** »
- ≡ Un moyen d'y accéder
- ≡ Un moyen de décrire les flux de données (paramètres, retours)
- ≡ Un nom, des métadonnées descriptives, etc.

Il existe deux systèmes de sémantique que nous nommerons l'approche explicite et l'approche implicite.

Sémantique des web service

Approche explicite

L'idée est d'utiliser un format de description du service.

Ce format s'appelle **WSDL** pour Web Service Definition Language.

Il ne décrit pas lui-même la communication concrète, mais encapsule un format de description des échanges comme SOAP ou UDDI.

Il contient :

- ≡ une définition des messages et des types de données utilisés
- ≡ une définition des opérations
- ≡ un couplage (binding) entre le nom de l'opération et la manière d'y accéder
- ≡ une définition de l'adresse du service.

Sémantique des web service

WSDL

```

<?xml version="1.0" encoding="UTF-8"?>
<wSDL:definitions name="SpaceWarGame" targetNamespace=" http://www.example.org/SpaceWarGame/"
                  xmlns:soap=" http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:tns=" http://www.example.org/SpaceWarGame/"
                  xmlns:wSDL=" http://schemas.xmlsoap.org/wsdl/"
                  xmlns:xsd=" http://www.w3.org/2001/XMLSchema"
                  xmlns:xsd1=" http://www.example.com/SpaceWarGame">
  <wSDL:types>
    <xsd:schema xmlns:xsd=" http://www.w3.org/2001/XMLSchema">
      <xsd:import namespace=" http://www.example.com/SpaceWarGame" schemaLocation="SpaceWarGame.xsd" />
    </xsd:schema>
  </wSDL:types>
  <wSDL:message name="getResultSheetResponse">
    <wSDL:part element="xsd1:ResultSheet" name="resultSheet" />
  </wSDL:message>
  <wSDL:message name="getResultSheetRequest">
    <wSDL:part element="xsd1:GameInformation" name="gameInformation" />
  </wSDL:message>
  <wSDL:message name="submitCommandSheetRequest">
    <wSDL:part name="commandSheet" element="xsd1:CommandSheet"></wSDL:part>
  </wSDL:message>
  <wSDL:message name="submitCommandSheetResponse">
    <wSDL:part name="result" element="xsd1:SubmitCommandSheetResult"></wSDL:part>
  </wSDL:message>
  <wSDL:message name="getResultSheet_faultMsg">
    <wSDL:part name="fault" element="xsd1:GetResultSheetFault"></wSDL:part>
  </wSDL:message>

```

Sémantique des web service

WSDL

```
<wsdl:portType name="SpaceWarGame">
  <wsdl:operation name="getResultSheet">
    <wsdl:input message="tns:getResultSheetRequest" />
    <wsdl:output message="tns:getResultSheetResponse" />
    <wsdl:fault name="fault" message="tns:getResultSheet_faultMsg"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="submitCommandSheet">
    <wsdl:input message="tns:submitCommandSheetRequest"></wsdl:input>
    <wsdl:output message="tns:submitCommandSheetResponse"></wsdl:output>
  </wsdl:operation>
</wsdl:portType>
```

Sémantique des web service

WSDL

```
<wsdl:binding name="SpaceWarGameSOAP" type="tns:SpaceWarGame">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getResultSheet">
    <soap:operation soapAction="http://www.example.org/SpaceWarGame/getResultSheet"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="fault"><soap:fault use="literal" name="fault"/></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="submitCommandSheet">
    <soap:operation soapAction="http://www.example.org/SpaceWarGame/submitCommandSheet"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

Sémantique des web service

WSDL

```
<wsdl:service name="SpaceWarGame">
  <wsdl:port binding="tns:SpaceWarGameSOAP" name="SpaceWarGameSOAP">
    <soap:address location="http://www.example.org/" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Sémantique des web service

REST

Une autre approche, implicite, se base sur la notion d'agent et de ressource. Elle provient du WEB, l'idée étant d'associer une URL à un service. On parle d'architecture RestFull.

exemple :

```
http://example.org/ws/calculatrice/somme/4/3
```

A cette adresse se trouvera uniquement le nombre 7.

Sémantique des web service

REST

`http ://example.org/ws/`

`calculatrice/`

`somme/`

`4/3`

Quelques ressources

☰ SOAP :

http://www.w3schools.com/webservices/ws_soap_intro.asp

☰ WSDL :

http://www.w3schools.com/webservices/ws_wsdل_intro.asp

☰ rewriting avec apache :

<http://www.urlrewriting.fr/tutoriel-reecriture.htm>

☰ rewriting en PHP : <http://www.urlrewriting.fr/tutoriel-urlrewriting-sans-moteur-rewrite.htm>