

TP n°1 : Développement et test d'un service web SOAP en Java

Introduction

Ce TP a pour objectif de comprendre le fonctionnement des services web SOAP en Java. Il permet de voir comment créer un service web, le déployer sur une adresse locale, consulter son fichier WSDL et tester ses méthodes à l'aide d'un client SOAP. Le TP a été réalisé en suivant les étapes montrées dans la vidéo du cours.

Présentation des services web SOAP

Un service web SOAP permet à des applications de communiquer entre elles via le réseau. Les échanges se font sous forme de messages XML et utilisent généralement le protocole HTTP. SOAP est basé sur des règles strictes et repose sur un fichier WSDL qui décrit les méthodes disponibles, les paramètres d'entrée et les réponses.

Contrairement à REST, SOAP utilise uniquement le format XML et est plus structuré. Il est souvent utilisé dans les systèmes nécessitant une communication fiable et bien définie.

Déploiement du service web

Le service web a été déployé à l'aide de la classe Applications.java. Cette classe utilise l'API JAX-WS pour publier le service à l'adresse <http://localhost:8888/>. Une fois le programme exécuté, le service devient accessible via cette URL.

The screenshot shows the IntelliJ IDEA interface with the project 'servicesoapexample1' open. The 'src' directory contains three Java files: 'Applications.java', 'Etudiant.java', and 'Monserviceweb.java'. The 'Applications.java' file is selected and contains the following code:

```
import javax.xml.ws.Endpoint;
public class Applications {
    public static void main(String[] args) {
        System.out.println("Début de déploiement de mon service");
        String url = "http://localhost:8888/";
        Endpoint.publish(url, new Monserviceweb());
        System.out.println("Le service web est déployé");
    }
}
```

The 'Run' tool window at the bottom shows the output of the application's execution:

```
C:\Users\DELL\.jdks\corretto-1.8.0_482\bin\java.exe ...
Début de déploiement de mon service
Le service web est déployé
```

Implémentation du service web

Le service web est défini dans la classe Monserviceweb.java. Cette classe est annotée avec `@WebService` afin d'être reconnue comme un service SOAP. Elle contient plusieurs méthodes accessibles au client.

La méthode convertir permet d'effectuer une conversion simple.

La méthode somme permet de calculer la somme de deux nombres.

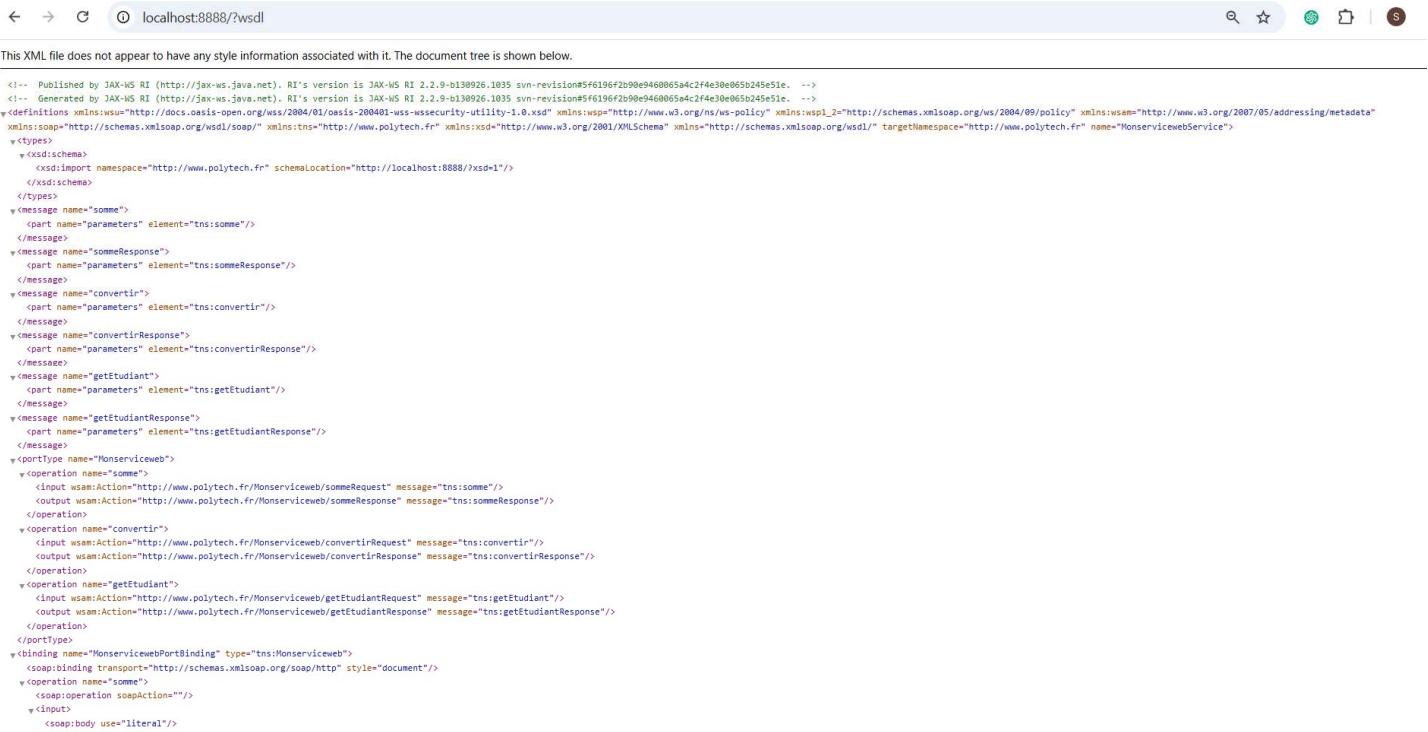
La méthode getEtudiant retourne un objet de type Etudiant.

Ces méthodes sont automatiquement décrites dans le fichier WSDL.

Description du WSDL

Le fichier WSDL est accessible via l'URL suivante : <http://localhost:8888/?wsdl>

Ce fichier décrit les opérations proposées par le service web, les types de données utilisés ainsi que les messages échangés entre le client et le serveur. Il permet au client de savoir comment appeler les méthodes du service.



```
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926-1035 svn-revision$5f0196f2b900e9460065a4c2f4e30e065b245e51e. -->
<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926-1035 svn-revision$5f0196f2b900e9460065a4c2f4e30e065b245e51e. -->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wspl="http://www.w3.org/ns/ws-policy" xmlns:wspl_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.polytech.fr/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://www.polytech.fr/" name="MonserviceWebService">
  <xsd:schema>
    <xsd:import namespace="http://www.polytech.fr/" schemaLocation="http://localhost:8888/?xsd=1"/>
  </xsd:schema>
  <types>
    <message name="somme">
      <part name="parameters" element="tns:somme"/>
    </message>
    <message name="sommeResponse">
      <part name="parameters" element="tns:sommeResponse"/>
    </message>
    <message name="convertir">
      <part name="parameters" element="tns:convertir"/>
    </message>
    <message name="convertirResponse">
      <part name="parameters" element="tns:convertirResponse"/>
    </message>
    <message name="getEtudiant">
      <part name="parameters" element="tns:getEtudiant"/>
    </message>
    <message name="getEtudiantResponse">
      <part name="parameters" element="tns:getEtudiantResponse"/>
    </message>
  </types>
  <portType name="Monserviceweb">
    <operation name="somme">
      <input wsam:Action="http://www.polytech.fr/Monserviceweb/sommeRequest" message="tns:somme"/>
      <output wsam:Action="http://www.polytech.fr/Monserviceweb/sommeResponse" message="tns:sommeResponse"/>
    </operation>
    <operation name="convertir">
      <input wsam:Action="http://www.polytech.fr/Monserviceweb/convertirRequest" message="tns:convertir"/>
      <output wsam:Action="http://www.polytech.fr/Monserviceweb/convertirResponse" message="tns:convertirResponse"/>
    </operation>
    <operation name="getEtudiant">
      <input wsam:Action="http://www.polytech.fr/Monserviceweb/getEtudiantRequest" message="tns:getEtudiant"/>
      <output wsam:Action="http://www.polytech.fr/Monserviceweb/getEtudiantResponse" message="tns:getEtudiantResponse"/>
    </operation>
  </portType>
  <binding name="MonservicewebPortBinding" type="tns:Monserviceweb">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operations>
      <operation name="somme">
        <soap:operation soapAction="" />
        <input>
          <soap:body use="literal"/>
        </input>
      </operation>
      <operation name="convertir">
        <soap:operation soapAction="" />
        <input>
          <soap:body use="literal"/>
        </input>
      </operation>
      <operation name="getEtudiant">
        <soap:operation soapAction="" />
        <input>
          <soap:body use="literal"/>
        </input>
      </operation>
    </operations>
  </binding>
</definitions>
```

Échange d'objets Java

Pour envoyer un objet Java via un service web SOAP, une classe spécifique est utilisée. La classe Etudiant est un Java Bean qui implémente l'interface Serializable et est annotée avec @XmlRootElement. Cela permet de transformer automatiquement l'objet Java en XML lors de l'échange.

La sérialisation permet donc d'envoyer l'objet du serveur vers le client sous forme de message SOAP.

Test du service avec SOAP UI

Le logiciel SOAP UI a été utilisé pour tester les méthodes du service web. À partir du WSDL, un projet SOAP a été créé. Chaque méthode a été testée en envoyant une requête SOAP et en observant la réponse du serveur.

Les tests ont permis de vérifier le bon fonctionnement des méthodes convertir, somme et getEtudiant.

The figure consists of three vertically stacked screenshots of the SoapUI application interface. Each screenshot shows a 'Request 1' tab with XML request and response panes.

- Top Screenshot (convertir method):**
 - Request XML (Raw):


```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
<soapenv:Header/>
<soapenv:Body>
<pol:convertir>
<arg0>1</arg0>
</pol:convertir>
</soapenv:Body>
</soapenv:Envelope>
```
 - Response XML (Raw):


```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:convertirResponse xmlns:ns2="http://www.polytech.fr">
<return>0.9</return>
</ns2:convertirResponse>
</S:Body>
</S:Envelope>
```
- Middle Screenshot (getEtudiant method):**
 - Request XML (Raw):


```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
<soapenv:Header/>
<soapenv:Body>
<pol:getEtudiant>
<arg0>1</arg0>
</pol:getEtudiant>
</soapenv:Body>
</soapenv:Envelope>
```
 - Response XML (Raw):


```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:getEtudiantResponse xmlns:ns2="http://www.polytech.fr">
<return>
<identifiant>1</identifiant>
<moyenne>19.0</moyenne>
<nom>Mario</nom>
</return>
</ns2:getEtudiantResponse>
</S:Body>
</S:Envelope>
```
- Bottom Screenshot (somme method):**
 - Request XML (Raw):


```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope">
<soapenv:Header/>
<soapenv:Body>
<pol:somme>
<parametre1>1</parametre1>
<arg1>4</arg1>
</pol:somme>
</soapenv:Body>
</soapenv:Envelope>
```
 - Response XML (Raw):


```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:sommeResponse xmlns:ns2="http://www.polytech.fr">
<return>6.0</return>
</ns2:sommeResponse>
</S:Body>
</S:Envelope>
```

Conclusion

Ce TP m'a permis de comprendre comment créer et déployer un service web SOAP en Java, consulter le fichier WSDL et tester les méthodes à l'aide d'un client SOAP. J'ai également appris comment échanger des objets Java entre un client et un serveur à l'aide de la sérialisation XML.