

Département Mathématiques Informatique

Génie du Logiciel et des Systèmes Informatiques Distribués

Rapport TP 2

Web services SOAP, WSDL, UDDI
avec JAXWS

Module Systèmes Distribués et Big Data Processing

Réalisé par :

HACHMI Mohamed Amine

Encadré par :

Pr. **YOUSSEFI** Mohamed

Sommaire

I. Création du Web service WSDL	3
1. Création de la classe BanqueService	3
2. Création de la classe Compte	4
II. Déploiement du service avec un serveur JaxWS	5
1. Création d'un simple serveur pour test du Web Service	5
2. Consultation du WSDL sur Browser HTTP	6
III. Test des opérations du web service avec SoapUI	7
1. Premier Test	7
2. Deuxième test : Sérialisation avec JAXB	8
IV. Création de clients SOAP	9
1. Client Java SOAP	9

I. Création du Web service WSDL

1. Création de la classe BanqueService

Ajout de la dépendance JaxWS au pom.xml :

```
<dependencies>
  <dependency>
    <groupId>com.sun.xml.ws</groupId>
    <artifactId>jaxws-ri</artifactId>
    <version>4.0.0</version>
    <type>pom</type>
  </dependency>
</dependencies>
```

```
BanqueService.java x
1   D:\ENSET GLSID\Cours 3ème Année\Microservices\tp2-webservice\src\main\java\bankWS\BanqueService.java
2
3   import jakarta.jws.WebMethod;
4   import jakarta.jws.WebParam;
5   import jakarta.jws.WebService;
6
7   import java.util.Date;
8   import java.util.List;
9
10  // POJO (Plain Old Java Object)
11  @WebService(serviceName = "BankWS")
12  public class BanqueService {
13      @WebMethod(operationName = "Convert")
14      public double conversion(@WebParam(name = "montant") double montant) {
15          return montant*10.54;
16      }
17      @WebMethod(operationName = "GetAccount")
18      public Compte getCompte(@WebParam(name = "code") int code) {
19          return new Compte(code, solde: Math.random()*9999, new Date());
20      }
21      @WebMethod(operationName = "ListAccounts")
22      public List<Compte> listeComptes() {
23          return List.of(
24              new Compte( code: 1, solde: Math.random()*9999, new Date()),
25              new Compte( code: 2, solde: Math.random()*9999, new Date()),
26              new Compte( code: 3, solde: Math.random()*9999, new Date())
27          );
28      }
29  }
```

2. Création de la classe Compte

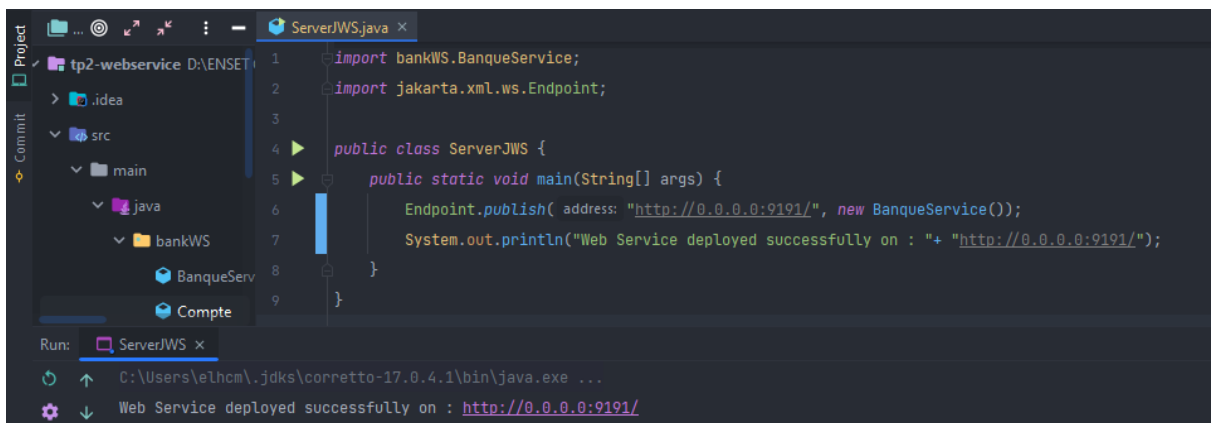
```
Compte.java x
1  package bankWS;
2
3  import java.util.Date;
4
5  public class Compte {
6      private int code;
7      private double solde;
8      private Date createdAt;
9
10     public Compte() {}
11
12     public Compte(int code, double solde, Date date) {
13         this.code = code;
14         this.solde = solde;
15         this.createdAt = date;
16     }
17
18     public int getCode() { return code; }
21
22     public void setCode(int code) { this.code = code; }
25
26     public double getSolde() { return solde; }
29
30     public void setSolde(double solde) { this.solde = solde; }
33
34     public Date getCreatedAt() { return createdAt; }
37
38     public void setCreatedAt(Date createdAt) { this.createdAt = createdAt; }
41 }
```

II. Déploiement du service avec un serveur JaxWS

1. Création d'un simple serveur pour test du Web Service

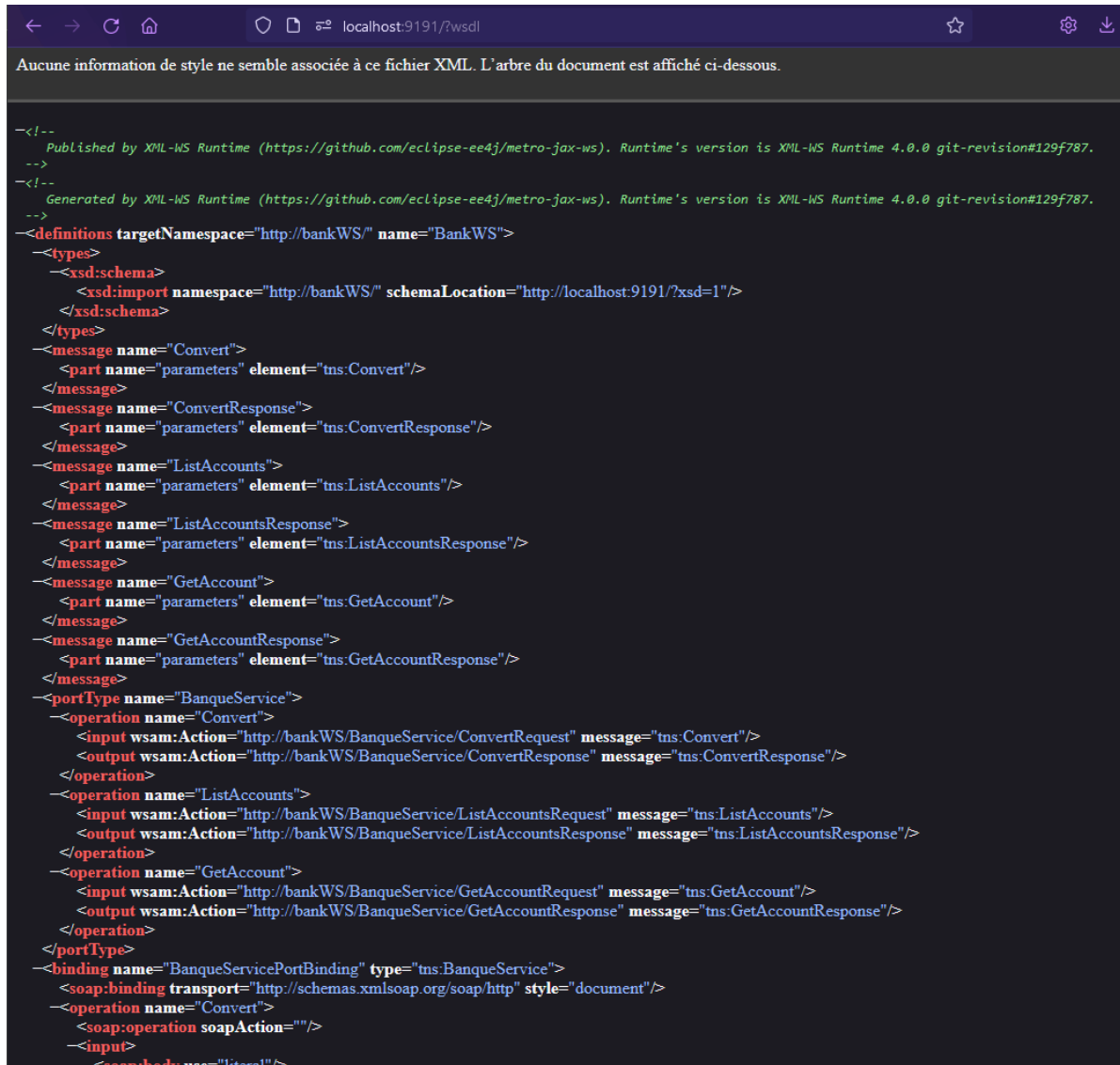
```
ServerJWS.java x
1  import bankWS.BanqueService;
2  import jakarta.xml.ws.Endpoint;
3
4  public class ServerJWS {
5  public static void main(String[] args) {
6      String address = "http://0.0.0.0:9191/";
7      Endpoint.publish(address, new BanqueService());
8      System.out.println("Web Service deployed successfully on : "+address);
9  }
10 }
```

Lancement du serveur :



```
Project
  tp2-webservice D:\ENSET\...
  .idea
  src
    main
      java
        bankWS
          BanqueServ
          Compte
Run: ServerJWS x
C:\Users\elham\.jdk\corretto-17.0.4.1\bin\java.exe ...
Web Service deployed successfully on : http://0.0.0.0:9191/
```

2. Consultation du WSDL sur Browser HTTP

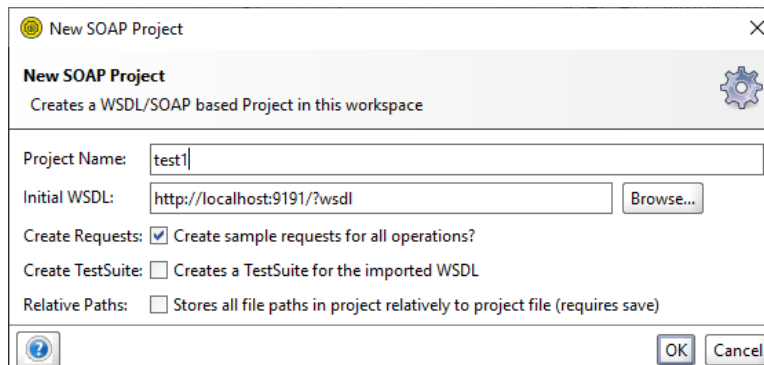


Aucune information de style ne semble associée à ce fichier XML. L'arbre du document est affiché ci-dessous.

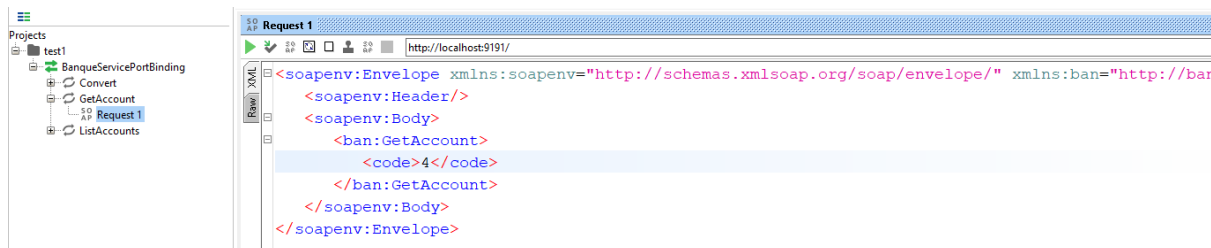
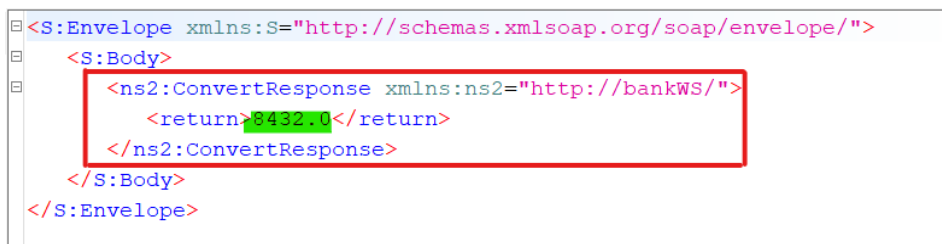
```
<!--
  Published by XML-WS Runtime (https://github.com/eclipse-ee4j/metro-jax-ws). Runtime's version is XML-WS Runtime 4.0.0 git-revision#129f787.
-->
<!--
  Generated by XML-WS Runtime (https://github.com/eclipse-ee4j/metro-jax-ws). Runtime's version is XML-WS Runtime 4.0.0 git-revision#129f787.
-->
<definitions targetNamespace="http://bankWS/" name="BankWS">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://bankWS/" schemaLocation="http://localhost:9191/?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="Convert">
    <part name="parameters" element="tns:Convert"/>
  </message>
  <message name="ConvertResponse">
    <part name="parameters" element="tns:ConvertResponse"/>
  </message>
  <message name="ListAccounts">
    <part name="parameters" element="tns:ListAccounts"/>
  </message>
  <message name="ListAccountsResponse">
    <part name="parameters" element="tns:ListAccountsResponse"/>
  </message>
  <message name="GetAccount">
    <part name="parameters" element="tns:GetAccount"/>
  </message>
  <message name="GetAccountResponse">
    <part name="parameters" element="tns:GetAccountResponse"/>
  </message>
  <portType name="BanqueService">
    <operation name="Convert">
      <input wsam:Action="http://bankWS/BanqueService/ConvertRequest" message="tns:Convert"/>
      <output wsam:Action="http://bankWS/BanqueService/ConvertResponse" message="tns:ConvertResponse"/>
    </operation>
    <operation name="ListAccounts">
      <input wsam:Action="http://bankWS/BanqueService/ListAccountsRequest" message="tns:ListAccounts"/>
      <output wsam:Action="http://bankWS/BanqueService/ListAccountsResponse" message="tns:ListAccountsResponse"/>
    </operation>
    <operation name="GetAccount">
      <input wsam:Action="http://bankWS/BanqueService/GetAccountRequest" message="tns:GetAccount"/>
      <output wsam:Action="http://bankWS/BanqueService/GetAccountResponse" message="tns:GetAccountResponse"/>
    </operation>
  </portType>
  <binding name="BanqueServicePortBinding" type="tns:BanqueService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="Convert">
      <soap:operation soapAction="">
        <input>
          <soap:body use="literal"/>
        </input>
      </soap:operation>
    </operation>
  </binding>
</definitions>
```

III. Test des opérations du web service avec SoapUI

1. Premier Test



Réponse :



Réponse :



2. Deuxième test : Sérialisation avec JAXB

```
package bankWS;

import jakarta.xml.bind.annotation.XmlAccessType;
import jakarta.xml.bind.annotation.XmlAccessorType;
import jakarta.xml.bind.annotation.XmlRootElement;
import jakarta.xml.bind.annotation.XmlTransient;

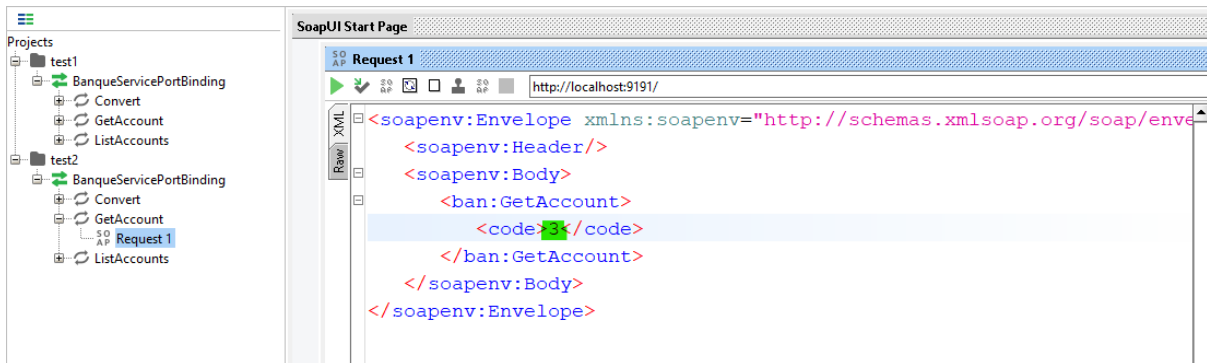
import java.util.Date;

@XmlRootElement(name="compte")
@XmlAccessorType(XmlAccessType.FIELD)
public class Compte {

    private int code;
    private double solde;
    @XmlTransient
    private Date createdAt;

    public Compte() {}

    public Compte(int code, double solde, Date date) {
        this.code = code;
        this.solde = solde;
        this.createdAt = date;
    }
}
```



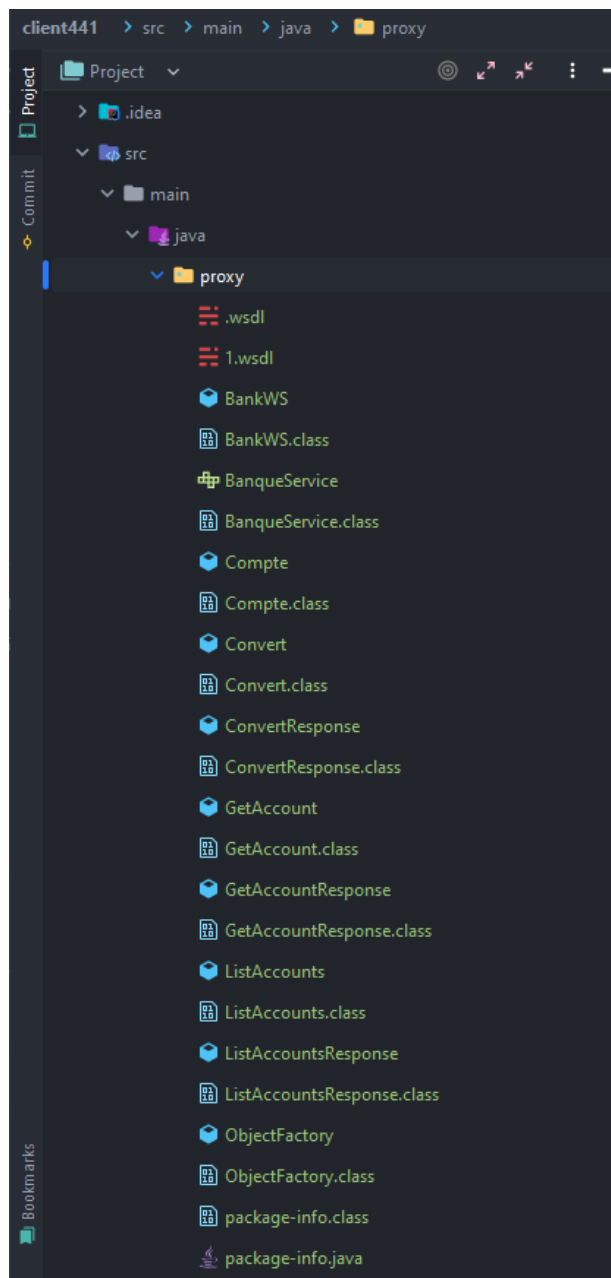
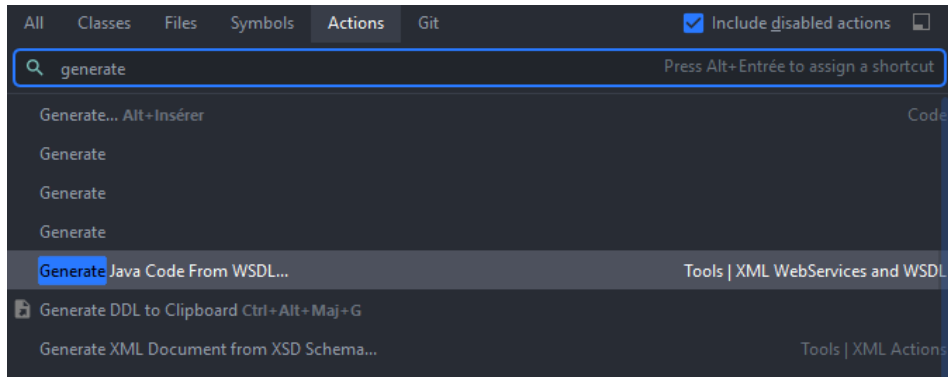
Réponse :

```
http://localhost:9191/

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:GetAccountResponse xmlns:ns2="http://bankWS/">
      <return>
        <code>3</code>
        <solde>9437.735820575319</solde>
      </return>
    </ns2:GetAccountResponse>
  </S:Body>
</S:Envelope>
```


IV. Création de clients SOAP

1. Client Java SOAP



```
ClientWS.java
3 import proxy.Compte;
4
5 public class ClientWS {
6     public static void main(String[] args) {
7         BanqueService stub = new BankWS().getBanqueServicePort();
8         System.out.println(stub.convert( montant: 9500));
9         Compte cp = stub.getAccount( code: 5);
10        System.out.println(cp.getCode());
11        System.out.println(cp.getSolde());
12    }
13 }
14
```

Run: ClientWS x

```
C:\Users\elhcm\.jdk\corretto-17.0.4.1\bin\java.exe ...
100129.999999999999
5
9156.964104983765
```

2. Client PHP SOAP

```
client.php
1 <?php
2 $client = new SoapClient( wsdl: "http://127.0.0.1:9191/BankWS?wsdl");
3 $param = new stdClass();
4 $param->montant = 450;
5 $req = $client -> __soapCall( name: "Convert", array($param));
6 $res = $req->return;
7 echo($res);
8
```

Run: client.php x

```
C:\xampp\php\php.exe "D:\ENSET GLSID\Cours 3ème Année\Microservices\tp2-webservice\phpClient\cli
4743
Process finished with exit code 0
```

Création d'un formulaire de conversion :

```
1 <?php
2     if(isset($_POST['montant'])) {
3         $montant = $_POST['montant'];
4         $client = new SoapClient( wsdl: "http://127.0.0.1:9191/BankWS?wsdl");
5         $param = new stdClass();
6         $param->montant = $montant;
7         $req = $client -> __soapCall( name: "Convert", array($param));
8         $res = $req->return;
9     }
10
11
12 <html>
13     <body>
14         <form method="post" action="client.php">
15             <h1>Conversion Euro - DH</h1>
16             <label for="montant">Montant en Euro:</label>
17             <input type="number" name="montant" id="montant" value="0" min="0" required />
18
19             <input type="submit" value="Convertir">
20         </form>
21         <br>
22         <h1>Résultat :</h1>
23         <?php
24             if(isset($res)) {
25                 echo("<b>".$res."</b>");
26             }
27         ?>
28     </body>
29 </html>
30
31
```

Conversion Euro - DH

Montant en Euro:

Résultat :

105.4