



RAPPORT DES TRAVAUX PRATIQUE

UNIVERSITÉ CHOUAIB DOUKKALI FACULTÉ DES
SCIENCES EL JADIDA

RISY OUSSAMA

Module du Web service et sémantique enseigné par
Dr. Omar Bouthoum

April 4, 2023

Sommaire

0.1	Introduction	3
0.2	Création d'un web service basé sur JAX-WS	3
0.2.1	Frais d'inscription	3
0.2.2	Les détails d'un cours sachant le code	3
0.2.3	Liste de cours	4
0.2.4	Ajouter un nouveau cours	4
0.3	Test avec SoapUI	4
0.3.1	Frais D'inscription	4
0.3.2	GetAllCours	5
0.3.3	Register	5
0.4	Création des clients	5
0.4.1	Client java	5
0.4.2	client JSP	8

Liste des figures

1	Code de la méthode Frais d'inscription	3
2	Code de la méthode getCours	3
3	Code de la méthode ListedeCours	4
4	Code de la méthode Registre	4
5	Test avec SoapUI de la méthode Frais d'inscription	4
6	Test avec SoapUI de la méthode GetAllCours	5
7	Test avec SoapUI de la méthode Register	5
8	Génération du STUB JAX-WS	6
9	Fichiers générés avec wisport	6
10	Code Client JWS	7
11	Test sur le serveur TOMCAT 8.5	7
12	Génération du proxy à partir du WSD	8
13	code java qui renvoie les frais d'inscrtion dans JSP	8
14	suite du code	9
15	Test 1 de client JSP sur le serveur	9
16	Test 2 de client JSP sur le serveur	9
17	Register code dans JSP	9
18	Test du code Register code sur le serveur	10
19	code client PHP	10
20	Caption	10

0.1 Introduction

Le présent rapport de TP décrit la conception et l'implémentation d'un web service Java basé sur JAX-WS pour permettre la consultation des frais d'inscription annuelle à une formation, des détails d'un cours en fonction de son code, ainsi qu'une liste de cours disponibles etc. Le rapport couvre également la phase de test avec l'outil SoapUI, ainsi que la création de différents clients pour accéder au web service, notamment un client Java, un client JEE (page JSP simple) et un client PHP. Le but de ce TP est de mettre en pratique les concepts de base de JAX-WS et de SOAP, ainsi que de comprendre comment développer et utiliser un web service pour fournir des fonctionnalités à des clients distants.

0.2 Création d'un web service basé sur JAX-WS

0.2.1 Frais d'inscription

La méthode "*fraisInsciCours*" prend en entrée un paramètre "*cd*" de type Long, qui représente le code d'un cours. La méthode utilise ensuite la méthode "*getOrDefault*" de l'objet "*COURS_prix*" pour récupérer le prix associé à ce code.

```
private static final Map<Long, Double> COURS_prix = new HashMap<>();
static {
    COURS_prix.put(123L, 1000.0);
    COURS_prix.put(456L, 800.0);
    COURS_prix.put(789L, 500.0);
}
@WebMethod(operationName="Frais_Dinscription")
public double frais_Insci_Cours(@WebParam(name="code")Long cd) {
    return COURS_prix.getOrDefault(cd, 0.0);
}
```

Figure 1: Code de la méthode Frais d'inscription

0.2.2 Les détails d'un cours sachant le code

Méthode qui permet de récupérer un objet "Cours" à partir de son code.

```
@WebMethod
public Cours getCours(@WebParam(name="code")Long cd) {
    List<Cours> listeCr = getAllCours();
    for (Cours c : listeCr) {
        if (c.getCode().equals(cd)) {
            return c;
        }
    }
    return null;
}
```

Figure 2: Code de la méthode getCours

0.2.3 Liste de cours

Méthode qui renvoie une liste d'objets "Cours".

```
@WebMethod
public List<Cours> getAllCours() {
    List<Cours> listeCr=new ArrayList<>();
    listeCr.add(new Cours(2L , "Big data",30));
    listeCr.add(new Cours (3L , "Data management",30));
    listeCr.add(new Cours(2L , "informatique décisionnelle",30));
    listeCr.add(new Cours (3L , "Programmation avancée",30));
    listeCr.add(new Cours(2L , "Théorie de l'information",30));
    listeCr.add(new Cours (3L , "Web Service",30));
    return listeCr;
}
```

Figure 3: Code de la méthode ListedeCours

0.2.4 Ajouter un nouveau cours

Méthode qui permet d'enregistrer un nouveau cours dans un registre.

```
@WebMethod
public boolean register(@WebParam(name = "code") Long code,
    @WebParam(name = "nom_cours") String nomCours,
    @WebParam(name = "total_ins") Integer totalInscriptions,
    @WebParam(name = "temps") String temps) {
    Registre registre = new Registre(code, nomCours, totalInscriptions, temps);
    /**/
    return true;
}
```

Figure 4: Code de la méthode Registre

0.3 Test avec SoapUI

0.3.1 Frais D'inscription

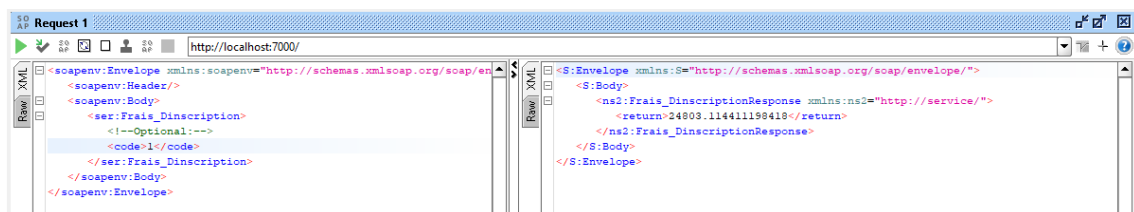


Figure 5: Test avec SoapUI de la méthode Frais d'inscription

0.3.2 GetAllCours

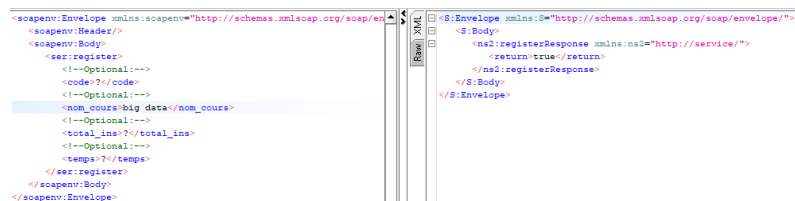


```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:getAllCours/>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getAllCoursResponse xmlns:ns2="http://service/">
      <return>
        <code>2</code>
        <nom_cours>Big data</nom_cours>
        <total_ins>30</total_ins>
      </return>
      <return>
        <code>3</code>
        <nom_cours>Data management</nom_cours>
        <total_ins>30</total_ins>
      </return>
      <return>
        <code>2</code>
        <nom_cours>informatique décisionnelle</nom_cours>
        <total_ins>30</total_ins>
      </return>
      <return>
        <code>3</code>
        <nom_cours>Programmation avancée</nom_cours>
        <total_ins>30</total_ins>
      </return>
      <return>
        <code>2</code>
        <nom_cours>Théorie de l'information</nom_cours>
        <total_ins>30</total_ins>
      </return>
      <return>
        <code>3</code>
        <nom_cours>Web Service</nom_cours>
        <total_ins>30</total_ins>
      </return>
    </ns2:getAllCoursResponse>
  </S:Body>
</S:Envelope>
```

Figure 6: Test avec SoapUI de la méthode GetAllCours

0.3.3 Register



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ser:register>
      <!--Optional:-->
      <code?></code>
      <!--Optional:-->
      <nom_cours>big data</nom_cours>
      <!--Optional:-->
      <total_ins?></total_ins>
      <!--Optional:-->
      <temps?></temps>
    </ser:register>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:registerResponse xmlns:ns2="http://service/">
      <return>true</return>
    </ns2:registerResponse>
  </S:Body>
</S:Envelope>
```

Figure 7: Test avec SoapUI de la méthode Register

0.4 Création des clients

0.4.1 Client java

Génération du STUB JAX-WS

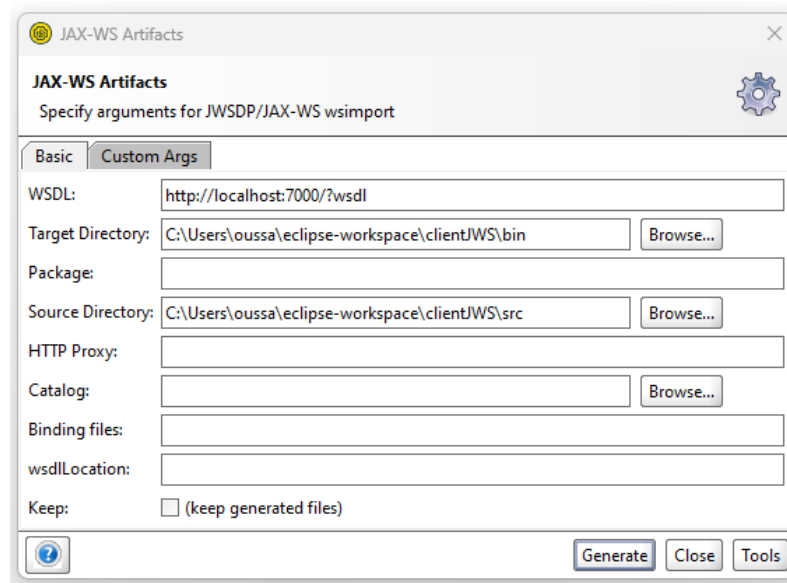


Figure 8: Génération du STUB JAX-WS

Fichiers générés

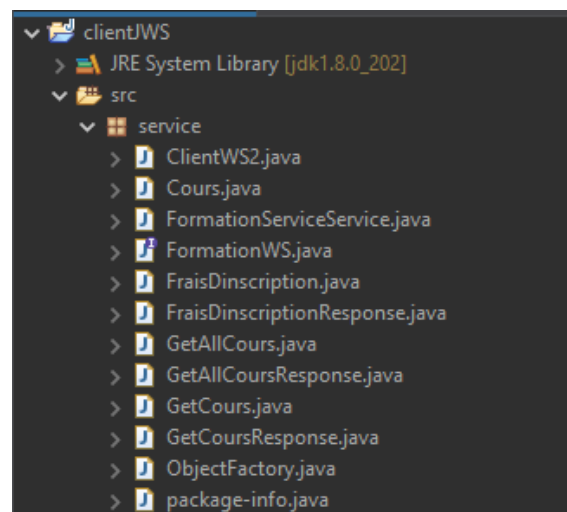


Figure 9: Fichiers générés avec wisport

```

package service;
import java.util.List;
public class ClientWS2 {

    public static void main(String[] args) {

        FormationWS stub = new FormationServiceService().getFormationWSPort();
        double res=stub.fraisDinscription((long) 123);
        System.out.println("les frais d'inscription annuelle"+'\u0020'+res+"DH");
        System.out.println("_____");
        Cours cr=stub.getCours((long) 31);
        System.out.println("le nombre d'inscrits au cours:"+cr.getNomCours()+"\u0020"+"est:"+cr.getTotalIns());
        System.out.println("_____");
        List<Cours> listeCr=stub.getAllCours();
        for(Cours c : listeCr) {
            System.out.println("Le nombre d'inscrits au cours:" +c.getNomCours()+"\u0020"+ "est:"+c.getTotalIns());
            System.out.println("_____");
        }
    }
}

```

Figure 10: Code Client JWS

```

<terminated> ClientWS2 [Java Application] C:\jdk1.8.0_202\bin\javaw.exe (Apr 2, 2023, 2:22:59 PM – 2:22:59 PM) [pid: 16924]
les frais d'inscription annuelle 22514.98047147985DH
_____
le nombre d'inscrits au cours:web service est:26
_____
Le nombre d'inscrits au cours:web semantique est:30
_____
Le nombre d'inscrits au cours:Big data est:45
_____

```

Figure 11: Test sur le serveur TOMCAT 8.5

0.4.2 client JSP

Génération du proxy à partir du WSDL

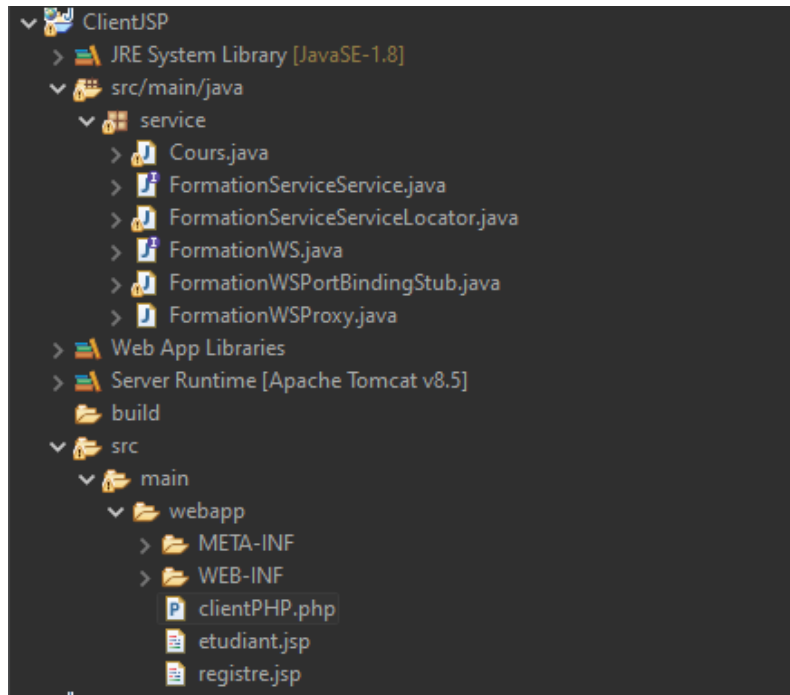


Figure 12: Génération du proxy à partir du WSD

Renvoie les frais d'inscription à partir du code

```
<%  
long cd=0;  
double res=0;  
if (request.getParameter("code")!=null) {  
  
    cd=Long.parseLong(request.getParameter("code"));  
    FormationWSPProxy service=new FormationWSPProxy();  
    res = service.frais_Dinscription(cd);  
}  
  
%>
```

Figure 13: code java qui renvoie les frais d'inscrtion dans JSP

```

<body>
<form method="Post" action="etudiant.jsp">
Code de la filière choisie :<input type="text" name="code" value="<%=cd%>">
<input type="submit" value="envoyer" name="btn">
</form>
<b>Frais d'inscription annuelles :<span style="color: #4CAF50;"><%= (int)res %> DH</span></b>
</body>
</html>

```

Figure 14: suite du code

Code de la filière choisie :
123

envoyer

Frais d'inscription annuelles :1000 DH

Figure 15: Test 1 de client JSP sur le serveur

Code de la filière choisie :
456

envoyer

Frais d'inscription annuelles :800 DH

Figure 16: Test 2 de client JSP sur le serveur

```

<% if (request.getMethod().equals("POST")) {
    Long code = Long.parseLong(request.getParameter("code"));
    String nomCours = request.getParameter("nom_cours");
    Integer totalInscriptions = Integer.parseInt(request.getParameter("total_ins"));
    String temps = request.getParameter("temps");

    FormationService service = new FormationService();
    boolean success = service.register(code, nomCours, totalInscriptions, temps);

    if (success) {
        %><h2>Registration successful!</h2><%
    } else {
        %><h2>Registration failed!</h2><%
    }
} %>

```

Figure 17: Register code dans JSP

Course Code:

Course Name:

Total Registrations:

Time:

Figure 18: Test du code Register code sur le serveur

```
<?php
$cd=0;
$res=0;
if (isset($_GET['code'])) {
    $cd=$_GET['code'];
    $client = new SoapClient('http://localhost:7000/FormationService?wsdl');
    $param = new stdClass();
    $param->code=$cd;
    $rep=$client->__soapCall("frais_Dinscription",array($param));
    $res=$rep->return;
}

?>

<html>
<body>
<form action="clientPHP.php">
code de la filière choisie : <input type="text" name="code" value="<?php echo($cd) ?>">
<input type="submit" value="envoyer">

</form>
Frais d'inscription annuelles : <b><?php echo ($res) ?></b>DH
</body>
</html>
```

Figure 19: code client PHP

code de la filière choisie :

Frais d'inscription annuelles : **1000DH**

Figure 20: Caption