

# **TP: TP Programmation Orienté Objets Distribuées**

Ayoub MOUSTAID - GLSID2  
Année universitaire : 2015/2016

# Cahier de Charge de Projet

---

On souhaite créer une application distribuée en utilisant les trois solutions : Sockets, RMI et Web Services. L'application permet de gérer un des appels téléphoniques. Chaque appel est défini par son numéro, sa date, sa durée, le nom de l'expéditeur et le nom du destinataire.

La partie serveur du projet se compose des parties suivantes:

- La couche métier contenant les éléments suivants
  - o Une classe Appel
  - o Une interface IOperateurMetier déclarant les opérations métier suivantes :
    - Ajouter un appel
    - Consulter tous les appels
    - Consulter les appels d'un expéditeur donné
  - o Une implémentation de cette interface qui suppose que les appels sont stockés dans une collection de type HashMap
- Une couche sockets qui représente un serveur multithread qui permet au client de consulter les appels d'un expéditeur donné.
- Une couche RMI qui contient :
  - o Un service RMI qui permet :
    - Consulter la moyenne des durées de tous les appels
    - Consulter les appels d'un expéditeur donné
  - o Un serveur RMI
- Une couche service SOAP qui contient :
  - o Un web service basé sur JaxWS qui permet de :
    - Ajouter un appel
    - Consulter les appels d'un expéditeur donné
  - o Un serveur JaxWS

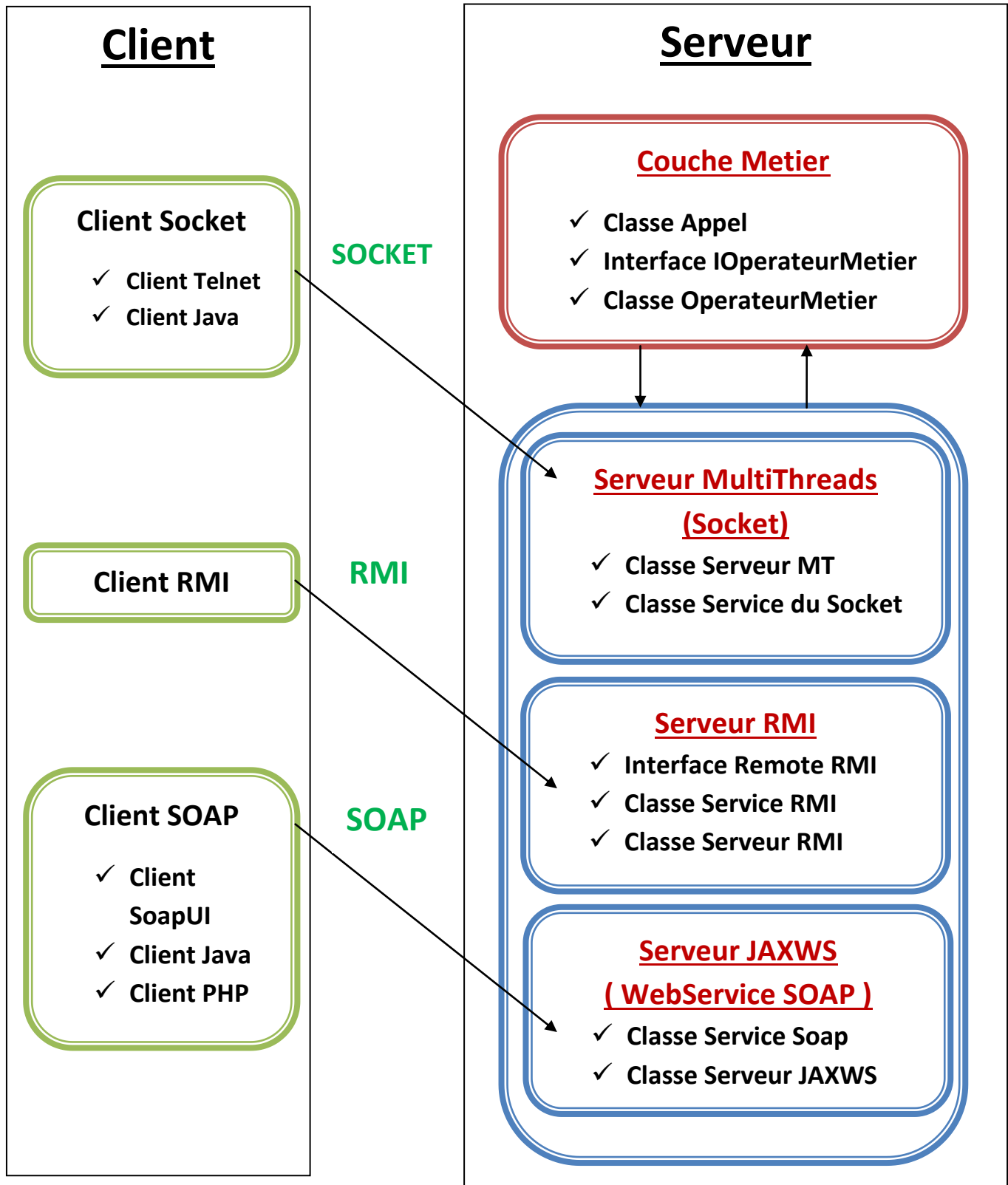
La partie cliente est représentée les type de clients suivants :

- Un client basé sur les sockets
- Un client RMI
- Un client SOAP (soapUI)
- Un client Soap Java
- Un Client Dot Net.
- Un client Soap PHP

## Sommaire

Cahier de Charge de Projet .....	2
Architecture Technique du projet.....	4
Partie Serveur .....	5
➤ <b>Structure</b> .....	5
➤ <b>Couche METIER</b> .....	6
➤ <b>Couche SOCKET</b> .....	10
➤ <b>Couche Service Soap</b> .....	15
Partie Client.....	17
➤ <b>Structure</b> .....	17
➤ <b>Client Socket</b> .....	18
➤ <b>Client RMI</b> .....	19
➤ <b>Client Web Service Java</b> .....	20
➤ <b>Client Web Service PHP</b> .....	21
Tests (Résultats) .....	24
➤ <b>Socket</b> .....	24
➤ <b>RMI</b> .....	25
➤ <b>WEB SERVICE</b> .....	25

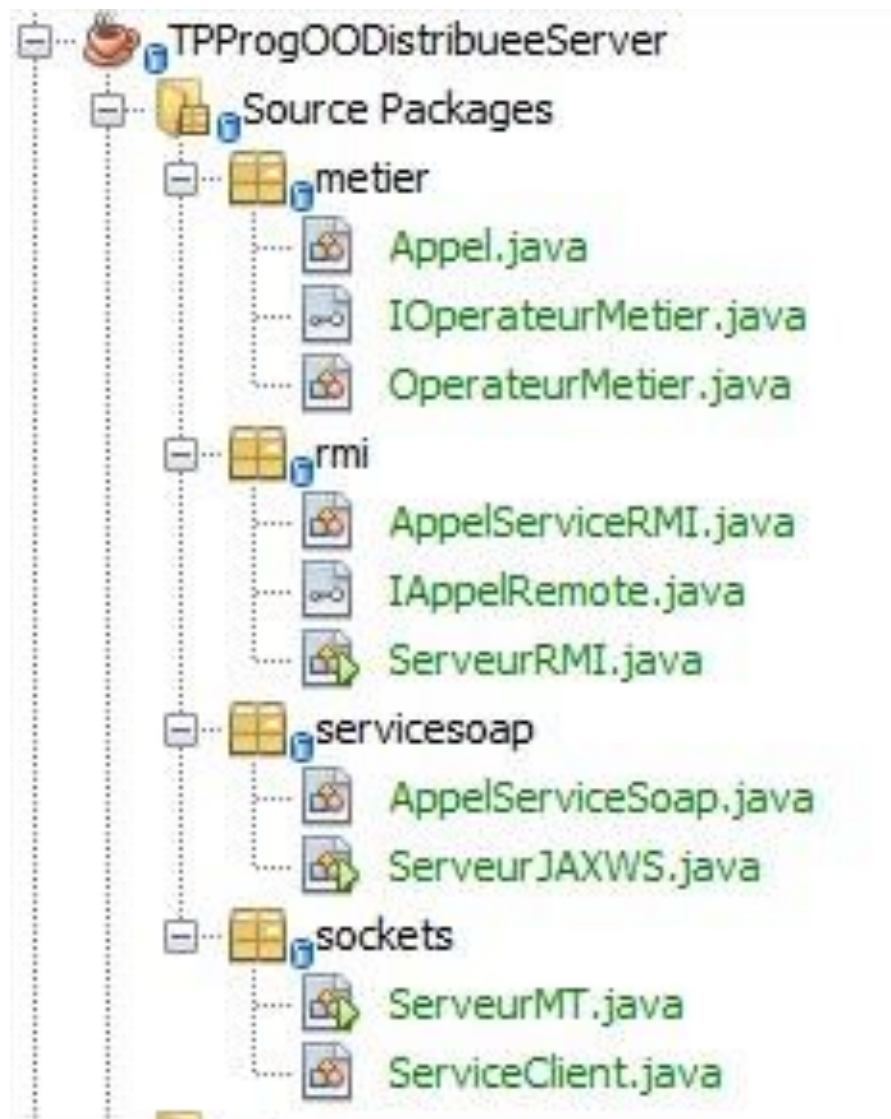
# Architecture Technique du projet



# Partie Serveur

---

## ➤ Structure



## ➤ Couche METIER

### ○ **Classe Appel**

```
@XmlRootElement
public class Appel implements Serializable {

    private int numero;
    private Date date;
    private double duree;
    private String expéditeur;
    private String destinataire;

    public Appel() {
    }
    public Appel(int numero, Date date, double duree, String expéditeur, String destinataire) {
        this.numero = numero;
        this.date = date;
        this.duree = duree;
        this.expéditeur = expéditeur;
        this.destinataire = destinataire;
    }
    public int getNumero() {
        return numero;
    }

    public void setNumero(int numero) {
        this.numero = numero;
    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    public double getDuree() {
        return duree;
    }

    public void setDuree(double duree) {
        this.duree = duree;
    }
}
```

```
public String getExpeditateur() {  
    return expeditateur;  
}  
  
public void setExpeditateur(String expeditateur) {  
    this.expeditateur = expeditateur;  
}  
  
public String getDestinataire() {  
    return destinataire;  
}  
  
public void setDestinataire(String destinataire) {  
    this.destinataire = destinataire;  
}  
}
```

- **Interface IOperateurMetier**

```
public interface IOperateurMetier {  
    void ajouterAppel(Appel appel);  
    Map<Integer,Appel> consulterAppels();  
    Map<Integer,Appel> consulterAppelsExpeditateur(String expeditateur);  
}
```

- **Classe OperateurMetier**

```
public class OperateurMetier implements IOperateurMetier {  
    private Map<Integer,Appel> appels=new HashMap();  
  
    public OperateurMetier() {  
        this.ajouterAppel(new Appel(1,new Date(),10,"Ayoub","Said"));  
        this.ajouterAppel(new Appel(2,new Date(),25,"Ayoub","Ahmed"));  
        this.ajouterAppel(new Appel(3,new Date(),20.5,"Soufiane","Karim"));  
        this.ajouterAppel(new Appel(4,new Date(),28,"Soufiane","Taha"));  
    }  
  
    @Override  
    public void ajouterAppel(Appel appel) {  
        appels.put(appel.getNumero(), appel);  
    }  
  
    @Override  
    public Map<Integer,Appel> consulterAppels() {  
        return appels;  
    }  
  
    @Override  
    public Map<Integer,Appel> consulterAppelsExpediteur(String expediteur) {
```



```

        Map<Integer,Appel> appelsOfExp=new HashMap();

        System.out.println("exp : "+expediteur);

        System.out.println("size : "+appels.size());

        for(Entry<Integer,Appel> e:appels.entrySet())
        {
            System.out.println("num
:"+e.getValue().getNumero()+",date:"+e.getValue().getDate()+",duree
:"+e.getValue().getDuree()+",destinataire
:"+e.getValue().getDestinataire()+",expediteur:"+e.getValue().getExpediteur()+"k
ey : "+e.getKey());

            if(e.getValue().getExpediteur().equals(expediteur))

                appelsOfExp.put(e.getKey(), e.getValue());

        }

        return appelsOfExp;

    }

}

```

## ➤ Couche SOCKET

### ○ **Classe ServiceClient**

```
public class ServiceClient extends Thread {  
    private Socket client;  
    private List<Socket> clients;  
    private IOperateurMetier operateurMetier;  
  
    public ServiceClient(Socket client,List<Socket> clients,IOperateurMetier  
operateurMetier) {  
        this.client = client;  
        this.clients=clients;  
        this.operateurMetier=operateurMetier;  
    }  
    @Override  
    public void run() {  
        try {  
            InputStream is = client.getInputStream();  
            BufferedReader br=new BufferedReader(new InputStreamReader(is));  
            OutputStream os=client.getOutputStream();  
            PrintWriter pw=new PrintWriter(os,true);  
            System.out.println("Connexion d'un client ");  
            pw.println("vous etes connecté");  
            while (true) {  
                String req=br.readLine();
```

```

        Map<Integer,Appel>
appels=opérateurMetier.consulterAppelsExpéditeur(req);

        if(appels.size()>0)
        {
            for(Entry<Integer,Appel> e:appels.entrySet())
            {
                for(Socket cl:clients)
                {
                    if(cl.isConnected())

                        new PrintWriter(cl.getOutputStream(),true).println("Numero
appel:"+e.getKey()+"-Expéditeur:"+e.getValue().getExpéditeur()+"-
Destinataire:"+e.getValue().getDestinataire()+"-
Durée:"+e.getValue().getDurée()+"-Date:"+e.getValue().getDate());

                } } }

            else
            {
                pw.println("pas d'appels pour cet expéditeur !");
            }
        }
    } catch (IOException ex) {

        Logger.getLogger(ServiceClient.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}
}

```

- **Classe ServeurMT**

```
public class ServeurMT extends Thread {

    List<Socket> clients = new ArrayList();

    IOperateurMetier operateurMetier = new OperateurMetier();

    @Override

    public void run() {

        try {

            ServerSocket ss = new ServerSocket(5453);

            System.out.println("waiting for connection...");

            while (true) {

                Socket s = ss.accept();

                clients.add(s);

                new ServiceClient(s, clients, operateurMetier).start();

            }

        } catch (IOException ex) {

            Logger.getLogger(ServeurMT.class.getName()).log(Level.SEVERE, null, ex);

        }

    }

    public static void main(String[] args) {

        new ServeurMT().start();

    }

}
```

➤ **Couche RMI**

○ **Interface IAppelRemote**

```
public interface IAppelRemote extends Remote {  
  
    double consulterMoyenneDurees() throws RemoteException;  
  
    Map<Integer,Appel> consulterAppelsExpediteur(String expéditeur) throws  
    RemoteException;  
  
}
```

○ **Classe AppelServiceRMI**

```
public class AppelServiceRMI extends UnicastRemoteObject implements  
IAppelRemote {  
  
    private IOperateurMetier operateurMetier=new OperateurMetier();  
  
    public AppelServiceRMI() throws RemoteException {  
  
        super() ; }  
  
    @Override  
  
    public double consulterMoyenneDurees() throws RemoteException{  
  
        double sumDurees=0;  
  
        for(Entry<Integer,Appel> e:operateurMetier.consulterAppels().entrySet())  
  
            sumDurees+=e.getValue().getDuree();  
  
        return sumDurees/operateurMetier.consulterAppels().size();  
  
    }  
  
    @Override  
  
    public Map<Integer, Appel> consulterAppelsExpediteur(String expéditeur)  
    throws RemoteException {  
  
        return operateurMetier.consulterAppelsExpediteur(expéditeur);  
  
    }  
  
}
```

- **Classe ServeurRMI**

```
public class ServeurRMI {  
    public static void main(String[] args) {  
        try {  
            LocateRegistry.createRegistry(1099);  
            AppelServiceRMI objetDistant=new AppelServiceRMI();  
            Naming.rebind("rmi://localhost:1099/AppelService", objetDistant);  
            System.out.println("ref : "+objetDistant);  
        } catch (Exception ex) {  
            Logger.getLogger(ServeurRMI.class.getName()).log(Level.SEVERE, null,  
ex);  
        }  
    }  
}
```

➤ **Couche Service Soap**

○ **Classe AppelServiceSoap**

```
@WebService(name="AppelWS")

public class AppelServiceSoap {

    private IOperateurMetier operateurMetier=new OperateurMetier();

    @WebMethod(operationName = "addAppel")

    public void ajouterAppel(@WebParam(name="appel")Appel
appel/*@WebParam(name="numero")int
numero,@WebParam(name="duree")double
duree,@WebParam(name="expediteur")String
expediteur,@WebParam(name="destinataire")String destinataire*/)

    {

        //Appel appel=new Appel(numero,new
Date(),duree,expediteur,destinataire);

        operateurMetier.ajouterAppel(appel);

    }

    @WebMethod(operationName="getAppelsByExp")

    public List<Appel>
consulterAppelsExpediteur(@WebParam(name="expediteur")String expediteur)

    {

        List<Appel> appels=new ArrayList();

        for(Entry<Integer,Appel>
e:operateurMetier.consulterAppelsExpediteur(expediteur).entrySet())

            appels.add(e.getValue());

        return appels;

    }
}
```

- **Classe ServeurJAXWS**

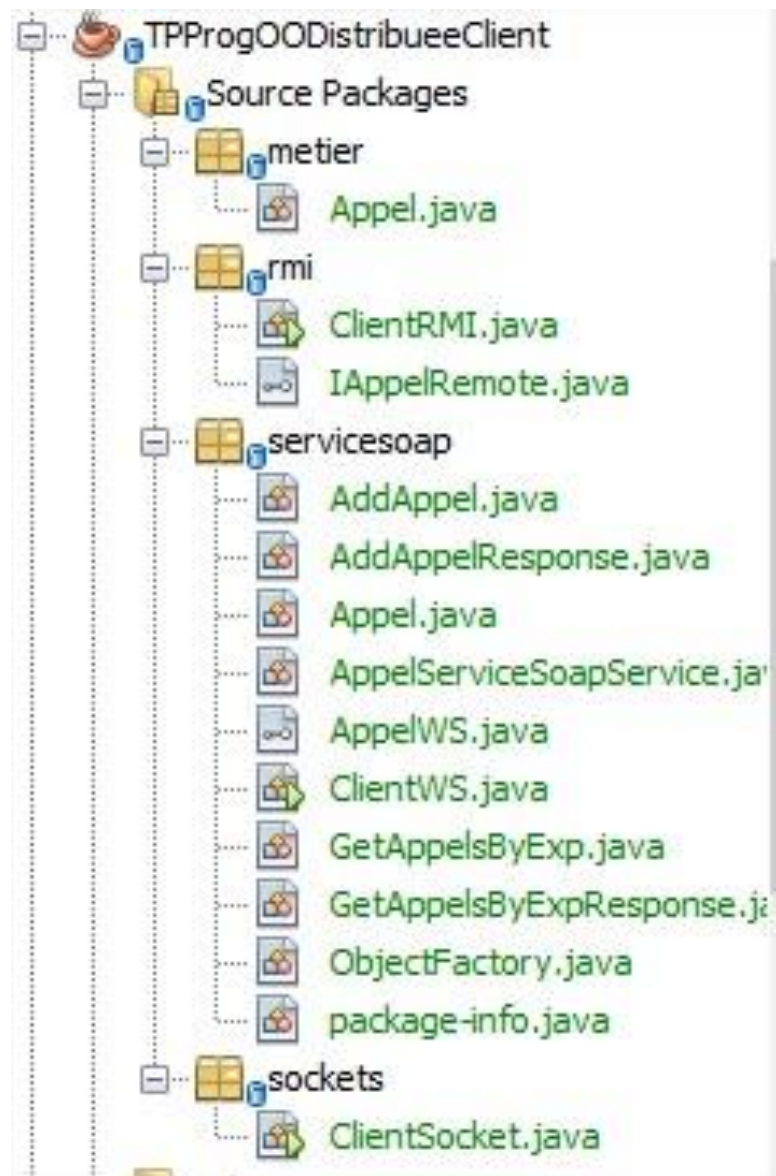
```
public class ServeurJAXWS {  
    public static void main(String[] args) {  
        String url="http://localhost:8585/";  
        Endpoint.publish(url, new AppelServiceSoap());  
        System.out.println("url = "+url);  
    }  
}
```



# Partie Client

---

## ➤ Structure



## ➤ Client Socket

```
public class ClientSocket {

    public static void main(String[] args) {

        try {

            Socket s=new Socket("localhost",5453);

            InputStream is=s.getInputStream();

            InputStreamReader isr=new InputStreamReader(is);

            BufferedReader br=new BufferedReader(isr);

            OutputStream os=s.getOutputStream();

            PrintWriter pw=new PrintWriter(os,true);

            Scanner clavier=new Scanner(System.in);

            String rep=br.readLine();

            System.out.println(rep);

            while (true) {

                System.out.println("Expeditateur : ");

                String text=clavier.next();

                pw.println(text);

                System.out.println("reponse : "+br.readLine());

            } }

        } catch (IOException ex) {

            Logger.getLogger(ClientSocket.class.getName()).log(Level.SEVERE, null,

ex);

        } }

    }
```

## ➤ Client RMI

```
public class ClientRMI {  
    public static void main(String[] args) {  
        try {  
            IAppelRemote  
stub=(IAppelRemote)Naming.lookup("rmi://localhost:1099/AppelService");  
            System.out.println("Total durees : "+stub.consulterMoyenneDurees());  
            System.out.println("Appels de - Ayoub ");  
            for(Entry<Integer,Appel>  
e:stub.consulterAppelsExpediteur("Ayoub").entrySet())  
                System.out.println("Numero appel:"+e.getKey()+"-  
Expediteur:"+e.getValue().getExpediteur()+"-  
Destinataire:"+e.getValue().getDestinataire()+"-  
Duree:"+e.getValue().getDuree()+"-Date:"+e.getValue().getDate());  
        } catch (Exception ex) {  
            Logger.getLogger(ClientRMI.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
}
```

## ➤ Client Web Service Java

```
public class ClientWS {  
    public static void main(String[] args) {  
        AppelWS stub=new AppelServiceSoapService().getAppelWSPort();  
        //test method d'ajout d'appel  
        Appel ap=new Appel();  
        ap.setNumero(20);  
        ap.setDate(new Date());  
        ap.setDuree(40);  
        ap.setExpediteur("Moustaid");  
        ap.setDestinataire("Ayoub");  
        stub.addAppel(ap);  
        //test method de consultation d'appels  
        List<Appel> appels=stub.getAppelsByExp("Moustaid");  
        System.out.println("Les appels de Moustaid");  
        for(Appel a:appels)  
        {  
            System.out.println("Numero appel:"+a.numero+"-  
Expediteur:"+a.expediteur+"-Destinataire:"+a.getDestinataire()+"-  
Duree:"+a.getDuree()+"-Date:"+a.getDate());  
        }  
    }  
}
```

➤ Client Web Service PHP

```
<?php
if(isset($_POST['btn']))
{
    $btn=$_POST['btn'];
    if($btn=='Ajouter')
    {
        $numero=$_POST['numero'];
        $duree=$_POST['duree'];
        $expediteur=$_POST['expediteur'];
        $destinataire=$_POST['destinataire'];
        $soapClient=new
SoapClient("http://localhost:8585/AppelService?wsdl");
        $params=new stdClass();
        $params->numero=$numero;
        $params->duree=$duree;
        $params->expediteur=$expediteur;
        $params->destinataire=$destinataire;
        $soapClient->__soapCall("addAppel",array($params));
    }
    elseif($btn=='Rechercher')
    {
        $search=$_POST['expediteur2'];
        $soapClient=new
SoapClient("http://localhost:8585/AppelService?wsdl");
```

```

        $params=new stdClass();

        $params->expediteur=$search;

        $resultat=$soapClient-
>__soapCall("getAppelsByExp",array($params));
    }
}
?>

<html>
<body>
<h1>Test Webservice avec Client PHP </h1>
<form action="appelclient.php" method="POST">
<h2>Ajouter un appel</h2>
Numero : <input type="text" name="numero"/><br/><br/>
Duree : <input type="text" name="duree"/><br/><br/>
Expediteur : <input type="text" name="expediteur"/><br/><br/>
Destinataire : <input type="text" name="destinataire"/>
<input name="btn" type="submit" value="Ajouter"/><br/>
<h2>Rechercher appel d'un Expediteur</h2>
Expediteur : <input type="text" name="expediteur2"/>
<input name="btn" type="submit" value="Rechercher"/>
<?php if(isset($resultat)){ ?>
<table border="1">
<tr>
<th>Numero</th>

```

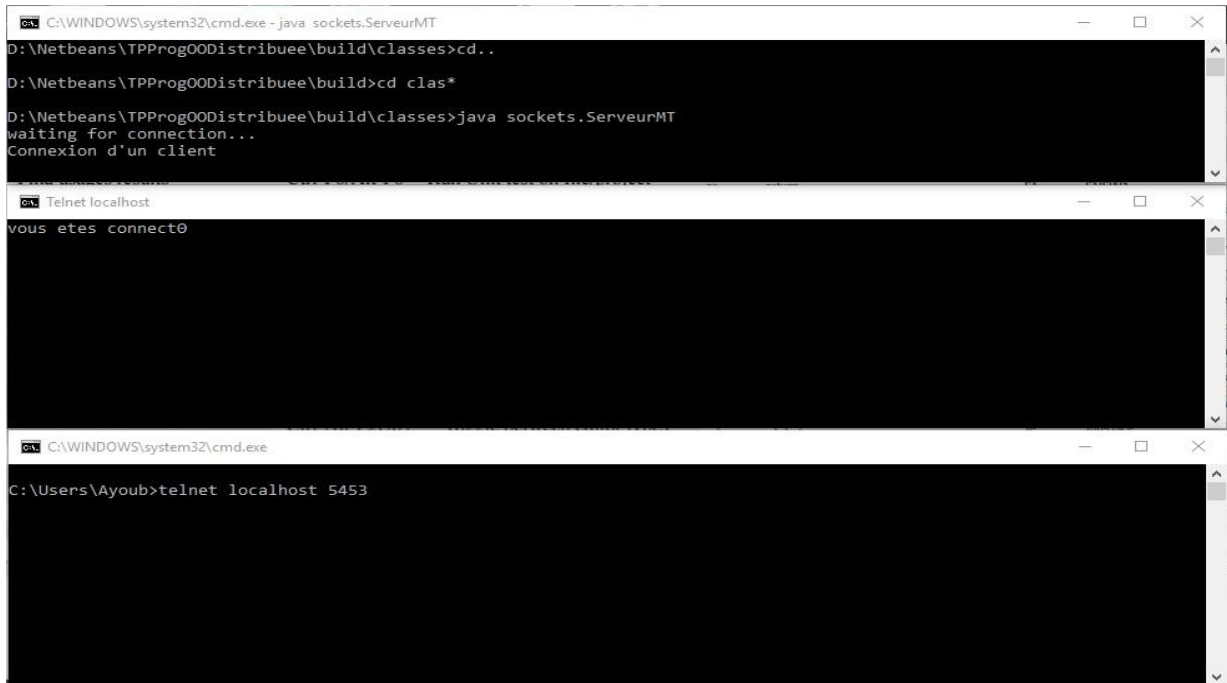
```
<th>Duree</th>
<th>Expediteur</th>
<th>Destinataire</th>
</tr>
<?php foreach($resultat->return as $appel) {?>
<tr>
<td><?php echo ($appel->numero) ?></td>
<td><?php echo ($appel->duree) ?></td>
<td><?php echo ($appel->expediteur) ?></td>
<td><?php echo ($appel->destinataire) ?></td>
</tr>
<?php } ?>
</table>
<?php } ?>

</form>
</body>
</html>
```

# Tests (Résultats)

## ➤ Socket

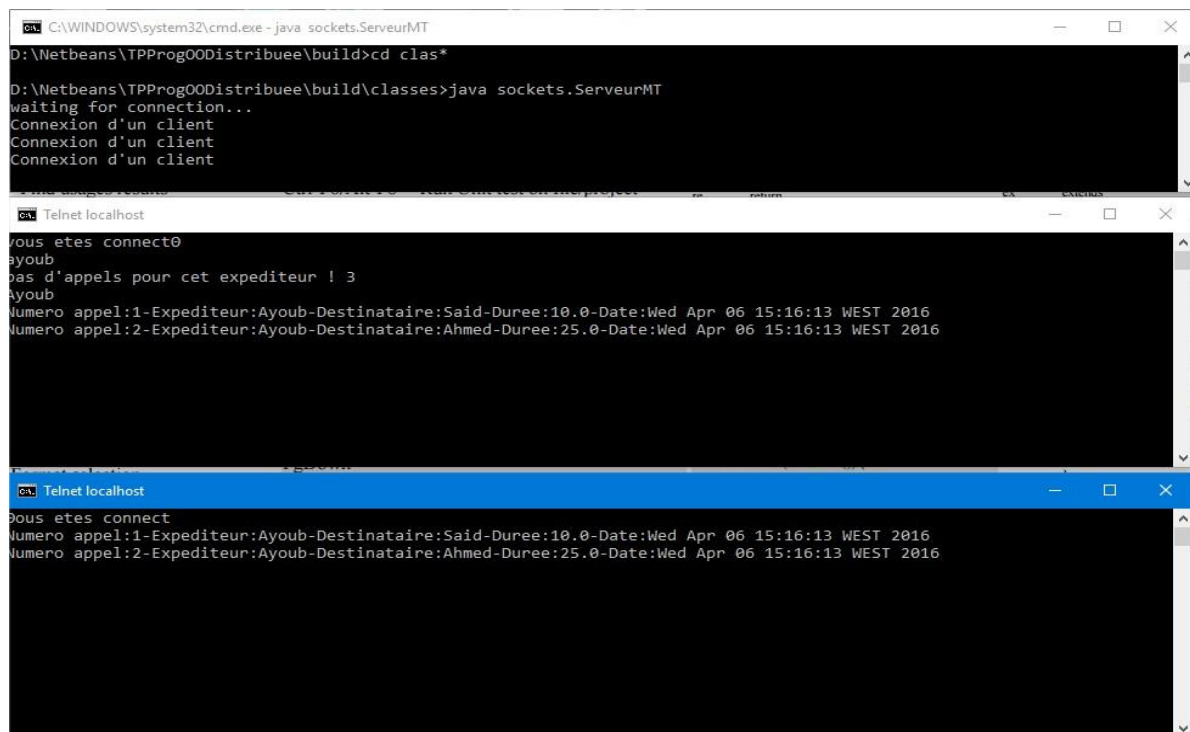
### Test client Telnet (lancement client et serveur)



```
C:\WINDOWS\system32\cmd.exe - java sockets.ServeurMT
D:\Netbeans\TPProgOODistribuee\build\classes>cd..
D:\Netbeans\TPProgOODistribuee\build>cd clas*
D:\Netbeans\TPProgOODistribuee\build\classes>java sockets.ServeurMT
waiting for connection...
Connexion d'un client

C:\WINDOWS\system32\cmd.exe
C:\Users\Ayoub>telnet localhost 5453
```

### Récupération liste d'appels d'un expéditeur pour tous les clients



```
C:\WINDOWS\system32\cmd.exe - java sockets.ServeurMT
D:\Netbeans\TPProgOODistribuee\build>cd clas*
D:\Netbeans\TPProgOODistribuee\build\classes>java sockets.ServeurMT
waiting for connection...
Connexion d'un client
Connexion d'un client
Connexion d'un client

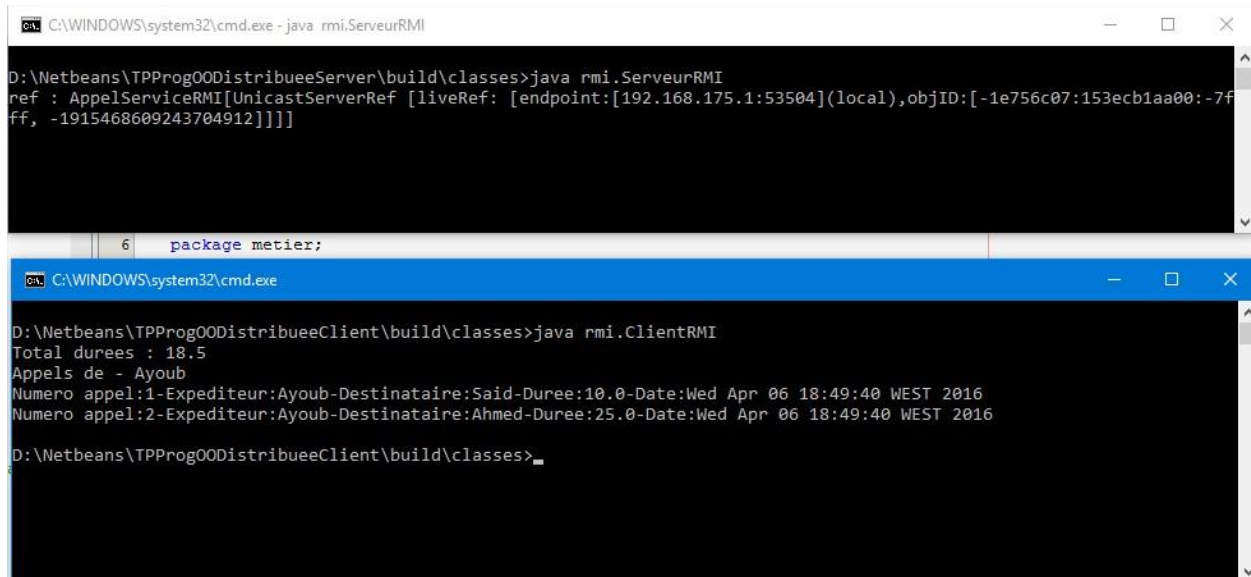
C:\WINDOWS\system32\cmd.exe
C:\Users\Ayoub>telnet localhost 5453
vous etes connect  
ayoub
pas d'appels pour cet exp  diteur ! 3
ayoub
Numero appel:1-Exp  diteur:Ayoub-Destinataire:Said-Duree:10.0-Date:Wed Apr 06 15:16:13 WEST 2016
Numero appel:2-Exp  diteur:Ayoub-Destinataire:Ahmed-Duree:25.0-Date:Wed Apr 06 15:16:13 WEST 2016

C:\WINDOWS\system32\cmd.exe
C:\Users\Ayoub>telnet localhost 5453
vous etes connect  
Numero appel:1-Exp  diteur:Ayoub-Destinataire:Said-Duree:10.0-Date:Wed Apr 06 15:16:13 WEST 2016
Numero appel:2-Exp  diteur:Ayoub-Destinataire:Ahmed-Duree:25.0-Date:Wed Apr 06 15:16:13 WEST 2016
```



## ➤ RMI

### Test RMI, Lancement serveur et client, ajout d'appel et affichage d'appels d'un expéditeur



The image shows two command prompt windows. The top window is titled 'C:\WINDOWS\system32\cmd.exe - java rmi.ServeurRMI' and shows the execution of 'java rmi.ServeurRMI' in the directory 'D:\Netbeans\TPProg00DistribueeServer\build\classes'. The output shows a reference to 'AppelServiceRMI[UnicastServerRef [liveRef: [endpoint:[192.168.175.1:53504](local),objID:[-1e756c07:153ecb1aa00:-7f...]]]]'. The bottom window is titled 'C:\WINDOWS\system32\cmd.exe' and shows the execution of 'java rmi.ClientRMI' in the directory 'D:\Netbeans\TPProg00DistribueeClient\build\classes'. The output shows 'Total durees : 18.5', 'Appels de - Ayoub', and two call details: 'Numero appel:1-Expéditeur:Ayoub-Destinataire:Said-Duree:10.0-Date:Wed Apr 06 18:49:40 WEST 2016' and 'Numero appel:2-Expéditeur:Ayoub-Destinataire:Ahmed-Duree:25.0-Date:Wed Apr 06 18:49:40 WEST 2016'.

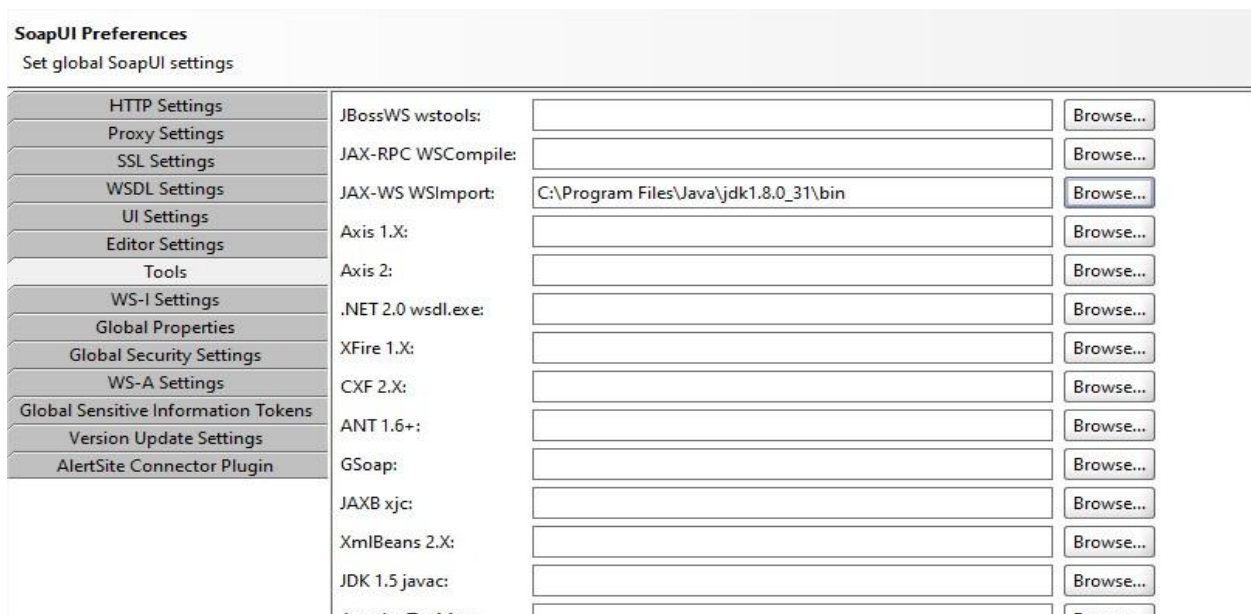
```
C:\WINDOWS\system32\cmd.exe - java rmi.ServeurRMI
D:\Netbeans\TPProg00DistribueeServer\build\classes>java rmi.ServeurRMI
ref : AppelServiceRMI[UnicastServerRef [liveRef: [endpoint:[192.168.175.1:53504](local),objID:[-1e756c07:153ecb1aa00:-7f...]]]]

C:\WINDOWS\system32\cmd.exe
D:\Netbeans\TPProg00DistribueeClient\build\classes>java rmi.ClientRMI
Total durees : 18.5
Appels de - Ayoub
Numero appel:1-Expéditeur:Ayoub-Destinataire:Said-Duree:10.0-Date:Wed Apr 06 18:49:40 WEST 2016
Numero appel:2-Expéditeur:Ayoub-Destinataire:Ahmed-Duree:25.0-Date:Wed Apr 06 18:49:40 WEST 2016
D:\Netbeans\TPProg00DistribueeClient\build\classes>
```

## ➤ WEB SERVICE

### Génération du proxy

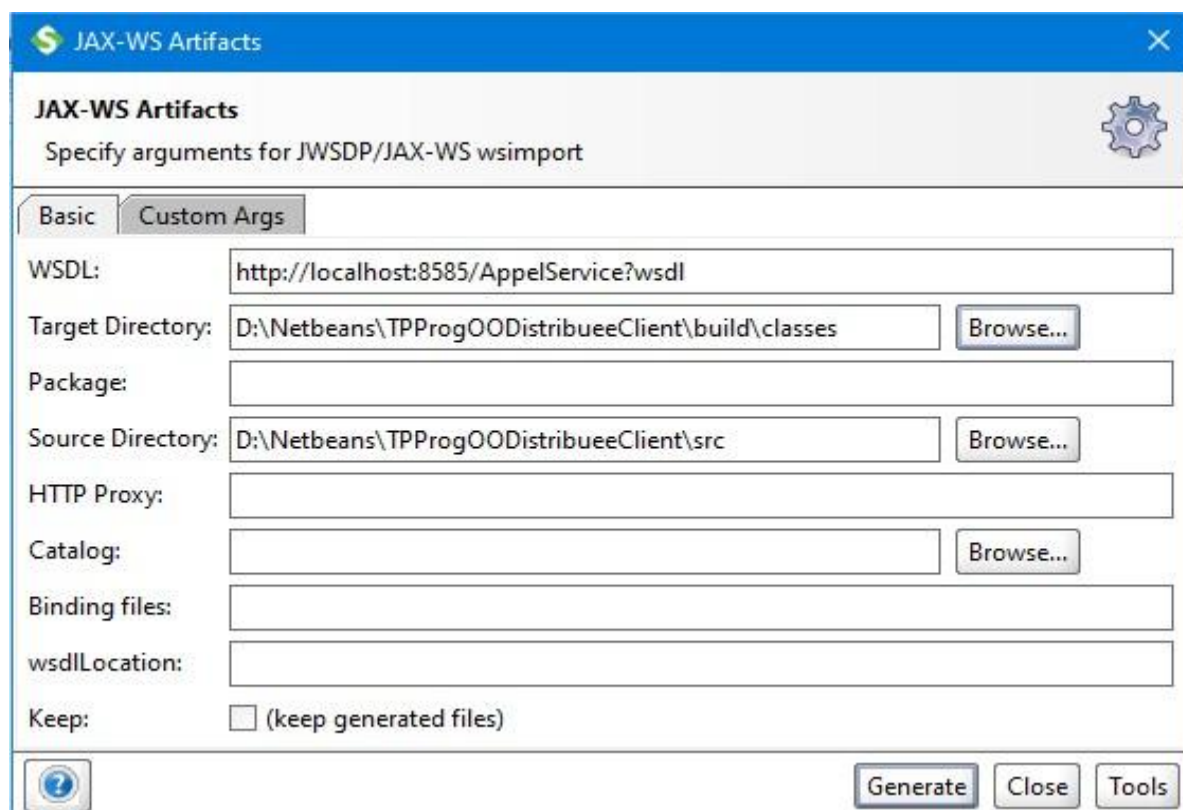
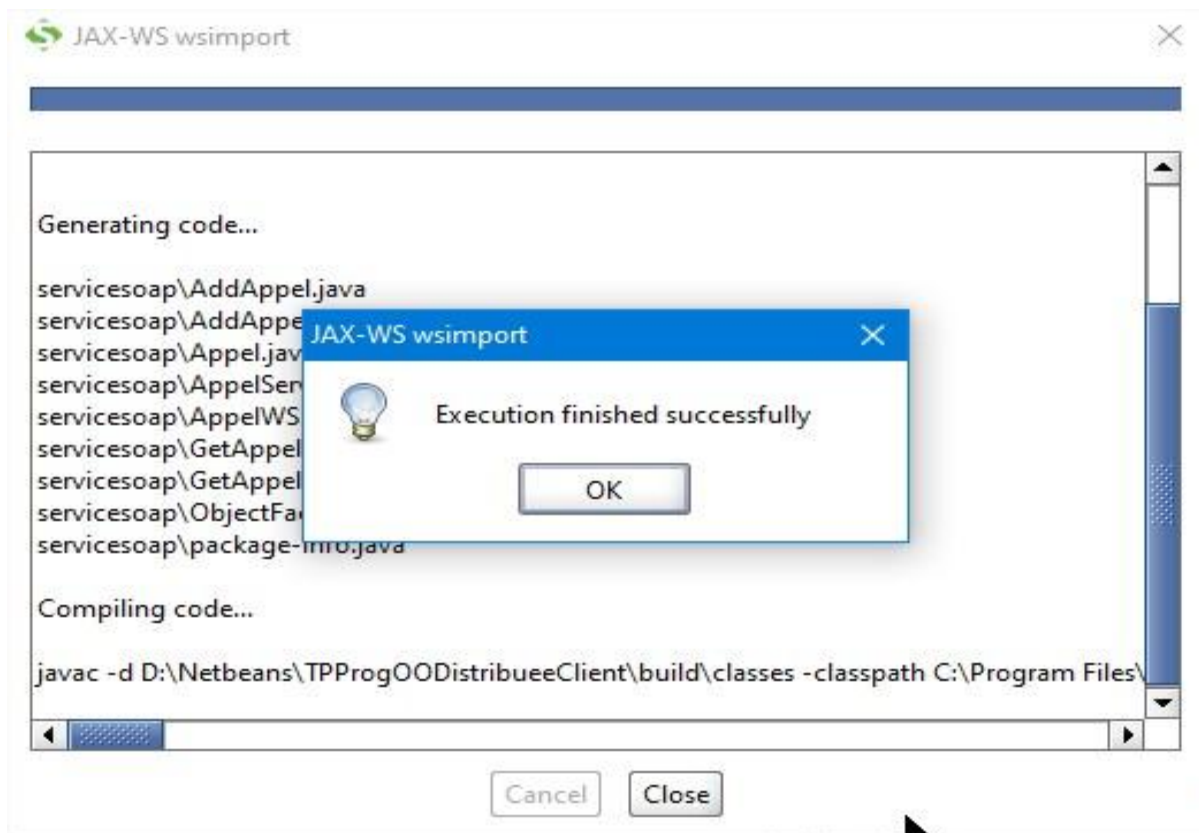
#### **\*Chemin de la commande WSIMPORT**



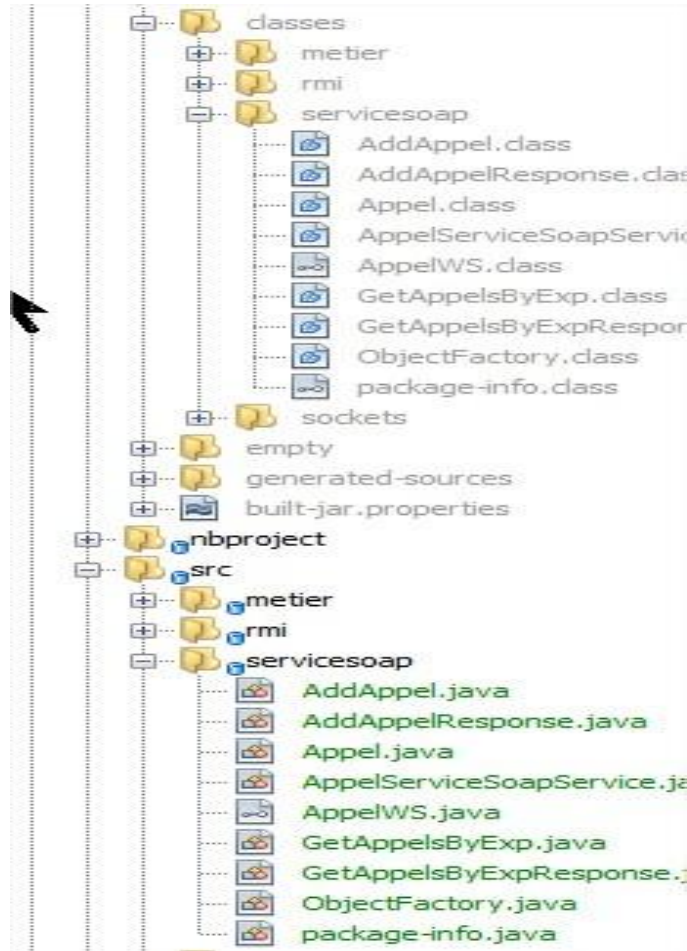
The image shows the 'SoapUI Preferences' dialog box. The 'Tools' tab is selected in the left sidebar. The main area lists various tools with their paths and 'Browse...' buttons. The 'JAX-WS WSIImport' path is set to 'C:\Program Files\Java\jdk1.8.0\_31\bin'.

Tool	Path	Action
JBossWS wstools:		Browse...
JAX-RPC WSCompile:		Browse...
JAX-WS WSIImport:	C:\Program Files\Java\jdk1.8.0_31\bin	Browse...
Axis 1.X:		Browse...
Axis 2:		Browse...
.NET 2.0 wsdl.exe:		Browse...
XFire 1.X:		Browse...
CXF 2.X:		Browse...
ANT 1.6+:		Browse...
GSoap:		Browse...
JAXB xjc:		Browse...
XmlBeans 2.X:		Browse...
JDK 1.5 javac:		Browse...
Apache TestMan...		Browse...

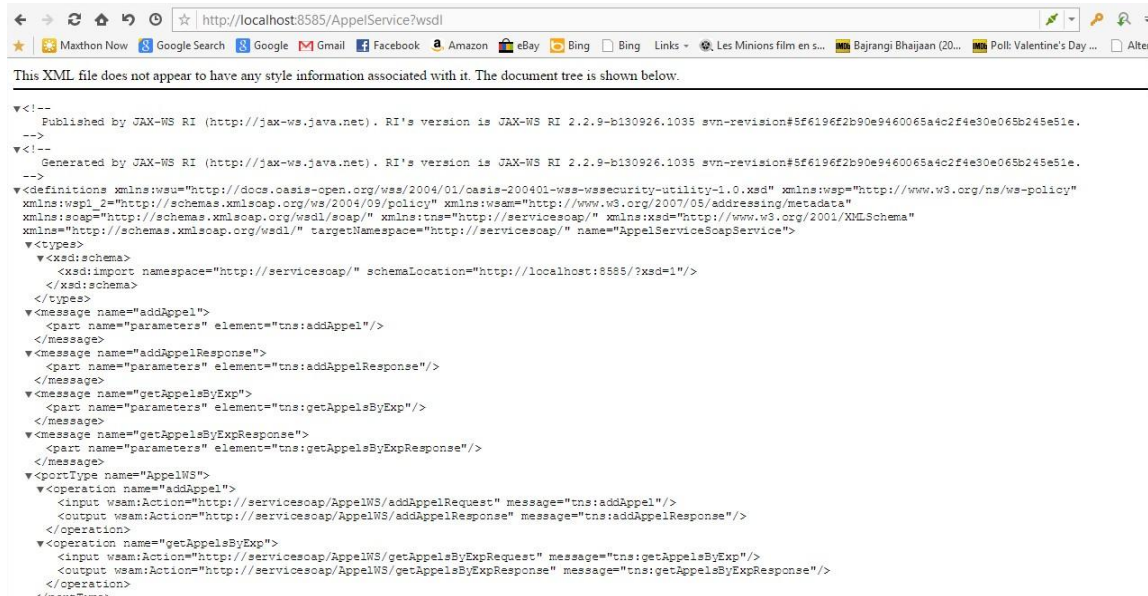
## \*Génération des fichiers



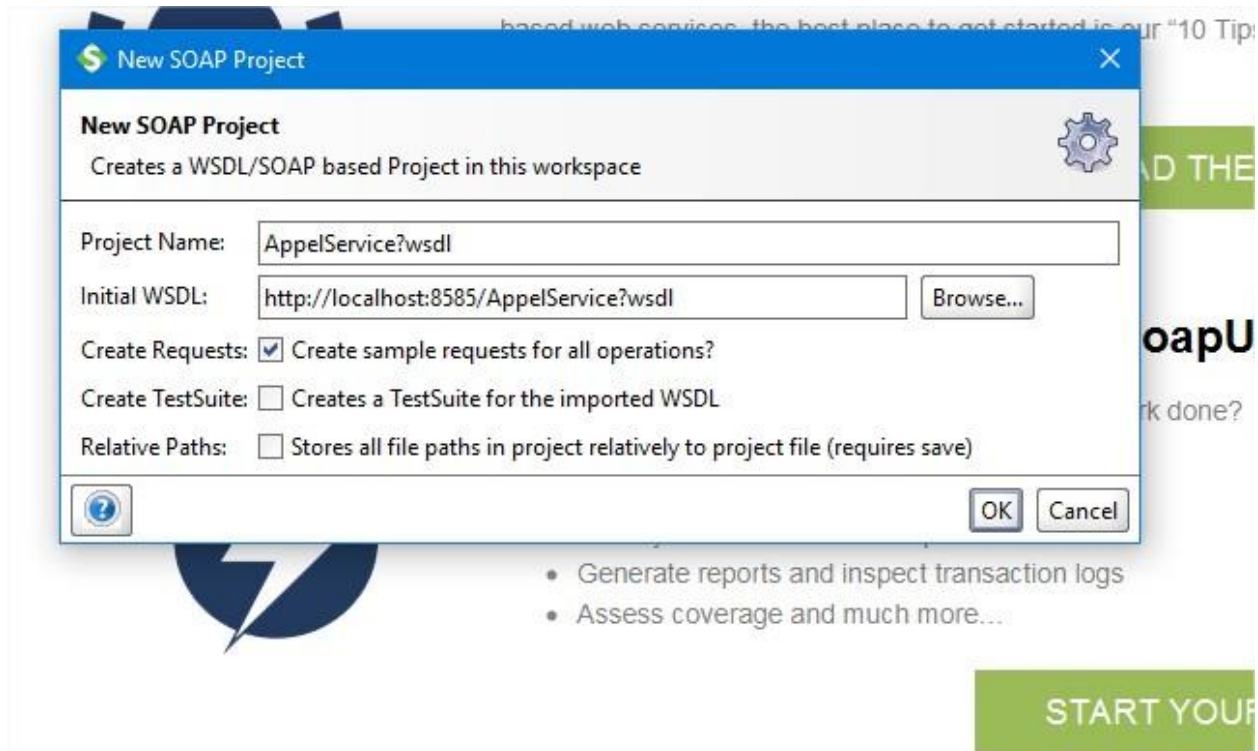
## \*Les fichiers Générés



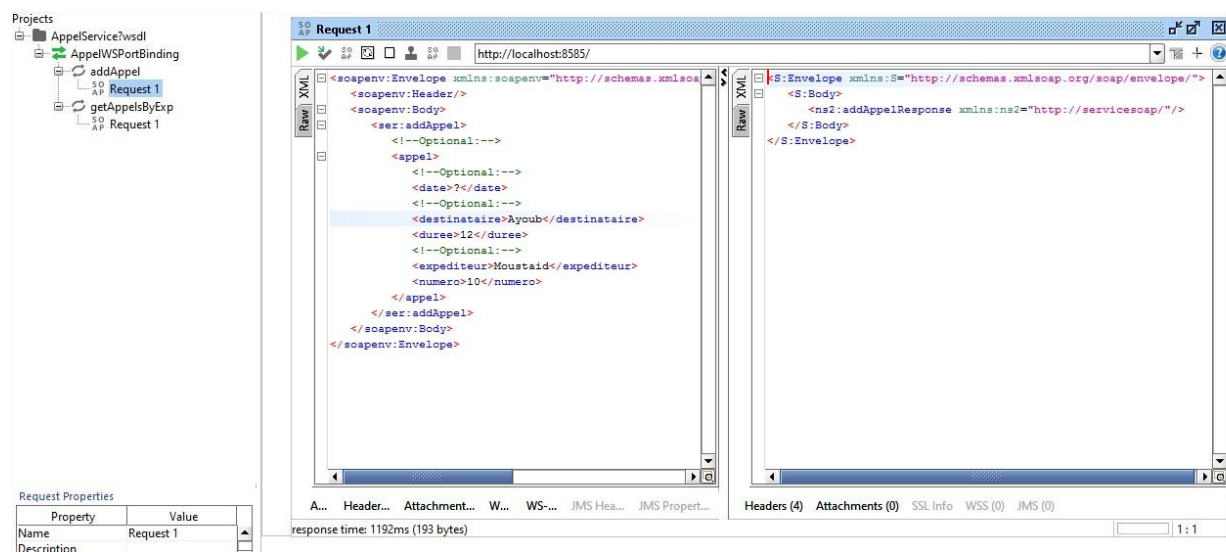
## \*Teste sur navigateur avec lien wsdl



**\*Lien WSDL**



### \*Test d'ajout d'appel



## \*Test Consultation appels d'un expéditeur

The screenshot shows the NetBeans IDE with a project named 'AppelService' and a sub-project 'AppelWSPortBinding'. The 'Request 1' tab is active, displaying a SOAP request and response. The request is a 'getAppelsByExp' call with an 'expediteur' of 'Moustaid'. The response is a 'getAppelsByExpResponse' with a 'destinataire' of 'Ayoub', a 'duree' of '12.0', and a 'numero' of '10'. The 'Request Properties' table at the bottom left shows the request name and value.

Property	Value
Name	Request 1

Request 1  
http://localhost:8585/  
response time: 84ms (327 bytes)

```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'>
  <soapenv:Header/>
  <soapenv:Body>
    <ser:getAppelsByExp>
      <!--Optional:-->
      <expediteur>Moustaid</expediteur>
    </ser:getAppelsByExp>
  </soapenv:Body>
</soapenv:Envelope>

<?xml version='1.0' encoding='UTF-8'>
<S:Envelope xmlns:S='http://schemas.xmlsoap.org/soap/envelope/'>
  <S:Body>
    <ns2:getAppelsByExpResponse xmlns:ns2='http://service/'>
      <return>
        <destinataire>Ayoub</destinataire>
        <duree>12.0</duree>
        <expediteur>Moustaid</expediteur>
        <numero>10</numero>
      </return>
    </ns2:getAppelsByExpResponse>
  </S:Body>
</S:Envelope>
```

## Test Client JAVA

### Lancement serveur

```
Netbeans - D:\Netbeans x TPProgOODistribueeServer (run) x

run:
url = http://localhost:8585/
```

### Test ajout est consultation d'appel

```
Netbeans - D:\Netbeans x TPProgOODistribueeServer (run) x TPProgOODistribueeClient (run) x

run:
Les appels de Moustaid
Numero appel:20-Expediteur:Moustaid-Destinataire:Ayoub-Duree:40.0-Date:Thu Apr 07 13:22:48 WEST 2016
BUILD SUCCESSFUL (total time: 3 seconds)
```



## Test Client PHP

←

→

↻

🏠

↶

🕒

☆

http://localhost/appelclient.php

★

☀️ Maxthon Now

🔍 Google Search

🔍 Google

📧 Gmail

📘 Facebook

📺 Amazon

# Test Webservice avec Client PHP

## Ajouter un appel

Numero :

Duree :

Expediteur :

Destinataire :

## Rechercher appel d'un Expediteur

Expediteur :

Numero	Duree	Expediteur	Destinataire
1	10	Ayoub	Said
2	25	Ayoub	Ahmed