

Séance 3 - Un premier programme RMI distribuable

- Lister tous les appels à des logiciels distants présents dans votre code.

```
Registry reg =  
    LocateRegistry.getRegistry("localhost"); /* se connecter à  
    l'annuaire de la machine */  
  
CompteurInterface c = (CompteurInterface)  
    reg.lookup("CompteurDistant");
```

- En utilisant les méthodes `.toString` des références obtenues par ces appels, retrouver les informations réseaux dans ces références.

```
CompteurInterface c = (CompteurInterface)  
    reg.lookup("CompteurDistant");  
  
System.out.println(c);
```

```
troha2u@iutnc-019-02:~/Documents/Programmation_Repartie/Seance2$ java  
Appel  
Proxy[CompteurInterface,RemoteObjectInvocationHandler[UnicastRef  
[liveRef:[endpoint:[100.64.80.50:44303](remote),objID:[-2877dd76:18d641e  
c52d:-7fff, -2832299956330613920]]]]]
```

- Que se passe-t-il quand vous choisissez le port du service au lieu de laisser le système d'exploitation le choisir ?
(voir la [documentation proposée dans ce cours](#)).

```
CompteurInterface rd = (CompteurInterface)  
    UnicastRemoteObject.exportObject(compteur, 334667);  
  
/* Un_port = un entier particulier ou 0 pour auto-assigné  
*/
```

```
troha2u@iutnc-019-02:~/Documents/Programmation_Repartie/Seance2$ javac  
*.java && java Bis  
REF SHEProxy[CompteurInterface,RemoteObjectInvocationHandler[UnicastRef  
[liveRef:[endpoint:[100.64.80.50:4667](remote),objID:[-3d08947f:18d6426c  
e05:-7fff, 7306549936068646135]]]]]
```

- D'ailleurs, quel est l'intérêt de laisser le système d'exploitation choisir le port du service pour vous ?

On est sûr qu'il soit libre d'utilisation

```
public class LancerCompteur{  
    •  
    •   public static void main (String args[]) throws RemoteException  
    {  
    •       Compteur compteur = new Compteur(); /* Créer une instance  
    de Compteur */  
    •       CompteurInterface rd = (CompteurInterface)  
    UnicastRemoteObject.exportObject(compteur,0);  
    •       /* Un_port = un entier particulier ou 0 pour auto-assigné  
    */  
    •       Registry reg = LocateRegistry.createRegistry(1099); /*  
    Création de l'annuaire */  
    •       reg.rebind("CompteurDistant", rd); /* Enregistrement de la  
    référence sous le nom "CompteurDistant" */  
    •   }  
    • }  
    •
```

Dans LancerCompteur.java on change la ligne qui crée le registre par une ligne pour le récupérer (créé préalablement dans un terminal dans le même répertoire que les fichiers relatifs au service via la commande rmiregistry).

```
Registry reg = LocateRegistry.createRegistry(1099);
```

devient donc :

```
Registry reg = LocateRegistry.getRegistry("localhost");
```

d'ailleurs, quel élément réseau/système permet de distinguer différents services hébergés sur une même machine ?

C'est le port qui permet de différencier les différents services (ex ssh, apache2).

Insérez dans votre compte-rendu les sources des programmes, des extraits d'exécutions, et l'arborescence de votre projet (avec la liste des fichiers par répertoire)

```
import java.util.Scanner;
import java.rmi.registry.Registry;
import java.rmi.registry.LocateRegistry;
import java.rmi.RemoteException;
import java.rmi.NotBoundException;
import java.rmi.server.ServerNotActiveException;
public class Appel{
    public static void main(String[] args) throws RemoteException,
NotBoundException, IndexOutOfBoundsException, ServerNotActiveException{

        Registry reg = LocateRegistry.getRegistry(args[0]); /* se
connecter à l'annuaire de la machine */

        ServiceDistributeur c = (ServiceDistributeur)
reg.lookup("LeDistributeur");
        System.out.println("REF SHE " + c);

        Scanner sc = new Scanner(System.in);
        String entree = sc.nextLine();
        while(!entree.equals("q")){
            if(entree.equals("n")){
                System.out.println("Valeur du compteur : " +
c.DonnerTicket());
                entree = sc.nextLine();
            }
        }
    }
}
```

```
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.rmi.server.ServerNotActiveException;

public interface ServiceDistributeur extends Remote{
    public int DonnerTicket() throws RemoteException,
ServerNotActiveException;
}
```

```
import java.rmi.Remote;
import java.rmi.RemoteException;
public class Distributeur implements ServiceDistributeur{
    private int num

    public Distributeur(int n){
        this.num = n;
    }

    public Distributeur(){
        this.num = 0;
    }

    public int DonnerTicket() throws RemoteException{
        this.num++;
        System.out.println(this.num); // Le service doit écrire sur son
terminal le numéro de ticket envoyé à chaque requête.
        return this.num;
    }
}
```

```
import java.rmi.server.UnicastRemoteObject;
import java.rmi.RemoteException;
import java.rmi.registry.Registry;
import java.rmi.registry.LocateRegistry;

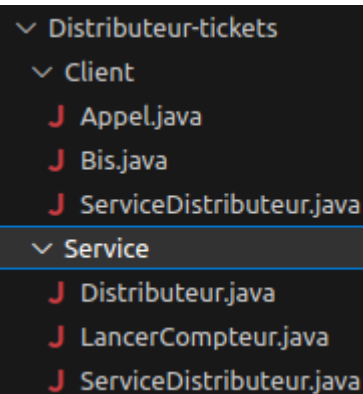
public class LancerCompteur{

    public static void main (String args[]) throws RemoteException {
        Distributeur compteur = new Distributeur(); /* Créer une
instance de Compteur */
        ServiceDistributeur rd = (DistributeurInterface)
UnicastRemoteObject.exportObject(compteur,0);
        /* Un_port = un entier particulier ou 0 pour auto-assigné */
        Registry reg = LocateRegistry.getRegistry("localhost"); /* se
connecter à l'annuaire de la machine */
        reg.rebind("LeDistributeur", rd); /* Enregistrement de la
référence sous le nom "CompteurDistant" */
    }
}
```

```
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.rmi.server.ServerNotActiveException;

public interface ServiceDistributeur extends Remote{
    public int DonnerTicket() throws RemoteException,
ServerNotActiveException;
}
```

```
troha2u@iutnc-019-02:~/Documents/Programmation_Repartie/Distributeur-tickets/Client$
javac *.java && java Appel 100.64.80.57
REF SHE Proxy[ServiceDistributeur,RemoteObjectInvocationHandler[UnicastRef [liveRef:
[endpoint:[100.64.80.57:34027](remote),objID:[36d54625:18d644ca1e1:-7fff, 41196366051
22783835]]]]]
n
Valeur du compteur : 70090016
n
Valeur du compteur : 70090017
```



Après avoir lancé notre service, nos camarades y ont fait appel, et le compteur s'est bel et bien incrémenté à chaque appel de leur part.

```
© 6C Ctroha2u@iutnc-125-04:~/Documents/Programmation_Repartie/Distributeur-tickets/Service$ java LancerCompteur
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
n
n
17
```

Concernant notre client, il a bien récupéré le compteur de nos camarades :

Stéphane Loppinet & Stanislas Troha RAIL 1

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
troha2u8iutnc-125-64:~/Documents/Programmation_Repartie/Distributeur-tickets/Client$ java Appel 100.64.80.227
REF SHE Proxy[ServiceDistributeur,RemoteObjectInvocationHandler[UnicastRef [LiveRef: [endpoint:[100.64.80.227:44637](remote),objID:[-3e86a1de:18d7eab14d9:-7fff, -5164940620843005272]]]]]
n
Valeur du compteur : 32203455
n
Valeur du compteur : 32203456
n
Valeur du compteur : 32203457
n
Valeur du compteur : 32203458
█
```