

1. **Interface Remote :** Tout objet que vous souhaitez rendre accessible via RMI doit implémenter l'interface `java.rmi.Remote` ou une de ses sous-interfaces. Cette interface sert de marqueur pour indiquer que l'objet peut être utilisé de manière distante.

```
java Copy code

import java.rmi.Remote;
import java.rmi.RemoteException;

public interface MyRemoteInterface extends Remote {
    String myRemoteMethod() throws RemoteException;
}
```

2. **Implémentation distante :** Vous fournissez ensuite une classe qui implémente cette interface. C'est cette classe qui contient les méthodes que vous souhaitez exposer de manière distante.

```
java Copy code

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class MyRemoteObject extends UnicastRemoteObject implements MyRemoteInterface {
    public MyRemoteObject() throws RemoteException {
        // Constructeur
    }

    public String myRemoteMethod() throws RemoteException {
        return "Hello from remote method!";
    }
}
```

3. **Enregistrement de l'objet distant :** Pour rendre l'objet distant accessible par d'autres machines, vous devez l'enregistrer auprès du registre RMI. Le registre RMI agit comme un annuaire pour les objets distants.

```
java Copy code

import java.rmi.Naming;

public class MyServer {
    public static void main(String[] args) {
        try {
            MyRemoteObject remoteObject = new MyRemoteObject();
            Naming.rebind("MyRemoteObject", remoteObject);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

4. **Invocation distante :** Les clients qui souhaitent utiliser l'objet distant peuvent récupérer une référence vers l'objet distant depuis le registre RMI et appeler ses méthodes comme s'il s'agissait d'objets locaux.

```
java Copy code

import java.rmi.Naming;

public class MyClient {
    public static void main(String[] args) {
        try {
            MyRemoteInterface remoteObject = (MyRemoteInterface) Naming
            String result = remoteObject.myRemoteMethod();
            System.out.println("Result from remote: " + result);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Etape 3 : il ne faut pas oublier de créer le registre

```
// Créer le registre RMI local sur le p
LocateRegistry.createRegistry(1099);
```