

# مثال RMI



د. فادي تركاوي

# Example



- نأخذ الآن مثال على .. RMI Programming ونبدأ بالمثال الشهير Hello World .
- نريد أن نكتب كائن يعمل في السيرفر يحتوي على دالة ترجع Hello ، ويقوم العميل باستدعاء هذه الدالة عن بعد..

# نبدأ بالبرمجة في جهة الخادم: Step1 in the Server



- سوف نحتاج أولاً إلى Interface يحتوي على جميع الدوال التي نريد العميل استدعائها وهنا فقط سوف تكون لدينا داله واحده وهي طباعه ..
- هذا الـ Interface يجب أن يقوم بعمل وراثه من الكلاس Remote حتى يقوم كلاس آخر (يمثل الكائن البعيد) بعمل تطبيق لهذا الأنترفيس .. اضافته الى أن الدوال التي سنكتبها في هذا الـ Interface يجب أن تقوم بعمل Throws النوع معين من الـ Exception هو RemoteException لأنه قد تحدث مشاكل اثناء استدعاء الداله مثلاً قطع الأتصال مع الخادم أو انهيار بالشبكه أو أيه مشكله أخرى .. لذلك جميع الدوال سوف تتعامل مع هذا الـ Exception.
- جميع الكلاسات الخاصه ببرمجه RMI موجوده في الباكج .. java.rmi لنرى الآن ملف Interface ولنطلق عليه: Hello

# Step 1 in the Server



```
// Hello Remote Object Interface
```

```
import java.rmi.Remote ;
```

```
import java.rmi.RemoteException ;
```

```
public interface Hello extends Remote
```

```
{
```

```
    public String getHelloMessage() throws RemoteException ;
```

```
}
```

## Step 2 in the Server



- أنتهت الخطوه الأولى ، الآن نقوم بكتابه كلاس يطبق هذا الـ `Interface` ، بالاضافه الى الوراثة من `UnicastRemoteObject` وهو الكلاس الخاص بأمر الـ `Marshalling` وارسال واستقبال البيانات ..وهو موجود في الباكج `java.rmi.server`

# Step 2 in the Server



```
// Hello Implementation
```

```
import java.rmi.RemoteException ;
```

```
import java.rmi.server.UnicastRemoteObject ;
```

```
public class HelloImpl extends UnicastRemoteObject implements Hello
```

```
{
```

```
    public HelloImpl () throws RemoteException
```

```
    { }
```

```
    public String getHelloMessage() throws RemoteException
```

```
    {
```

```
        return "Hello Distributed Computing" ;
```

```
    }
```

```
}
```

## Step 3 in the Server



- الخطوة الثالثة وهي كتابة السيرفر Server ، وهنا سوف نقوم بعمل كائن من HelloImpl وندخل هذا الكائن في الـ registry ونعطيه أي اسم ما، وبالطبع الكلاينت يجب أن يكون لديه هذا الأسم لكي يحصل على الـ reference فيما بعد ..

## Step 3 in the Server



- `Naming.rebind(objectName,myObject);`
  - هي التي تقوم باضافه سجل في ملف registry يحتوي على اسم الكائن وعنوانه . `Reference` طبعا هذه الخدمه Registry سوف نقوم بتشغيلها قبل أن يعمل السيرفر حتى تتم اضافة هذا السجل فيه .. يمكنك عن طريق هذه الخدمه أن تسجل كائن باسم ما ، ويقوم الكلاينت فيما بعد بالاتصال بهذه الخدمه بالأسم المعين ليحصل على الكائن .. طبعا الأضافه في ملف Registry تتم فقط من جهه السيرفر . الكلاينت يطلب من الـ Registry كائن عن طريق الأسم..
  - ملف `Server` ، لاحظ أنه بعد عمل `rebind` سوف نقوم بطباعه جمله تدل أن السيرفر يعمل الآن وفي حال اتصال

# Step 4 in the Server



```
// RMI Server
import java.rmi.RemoteException ;
import java.rmi.Naming ;
import java.net.MalformedURLException ;

public class Server
{
    final static String HOST = "localhost" ;

    public static void main (String args[])
    {
        try
        {
            HelloImpl myObject = new HelloImpl();
            String objectName = "rmi://" + HOST + "/MyHello" ;
            Naming.rebind(objectName,myObject);

            System.out.println("Binding Complete .....");
        }
        catch (RemoteException e)
        { e.printStackTrace(); }
        catch (MalformedURLException e)
        { e.printStackTrace(); }

    }
}
```

# Step 1 in Client



- كل ما علينا في جهة الكلاينت الحصول على reference للكائن ، ومن ثم استدعاء تلك الداله ، فقط .. ويتم الحصول على الكائن من خلال الداله  
lookUp الموجوده في الكلاس.. Naming
- Hello myObject = (Hello) Naming.lookup(objectName);

# Step 1 in Client



```
// RMI Client
import java.rmi.Naming ;
import java.rmi.RemoteException ;
import java.rmi.NotBoundException ;
import java.net.MalformedURLException ;
public class Client
{
    public final static String HOST = "localhost" ;

    public static void main (String args[])
    {
        try
        {
            String objectName = "rmi://" + HOST + "/MyHello" ;

            Hello myObject = (Hello) Naming.lookup(objectName);

            System.out.println(myObject.getHelloMessage());
        }
        catch (RemoteException e)
        { e.printStackTrace(); }
        catch (NotBoundException e)
        { e.printStackTrace(); }
        catch (MalformedURLException e)
        { e.printStackTrace(); }
    }
}
```

# Step 1 شرح البرنامج السابق



- كل ما علينا في جهة الكلاينت الحصول على reference للكائن ، ومن ثم استدعاء تلك الداله ، فقط .. ويتم الحصول على الكائن من خلال الداله `lookup` الموجوده في الكلاس `Naming`.
- `Hello myObject = (Hello) Naming.lookup(objectName);`
- الآن بعد الحصول على reference للكائن سوف يكون نوعه `Object` ، لذلك نقوم بعمل `cast` الى النوع `Hello` .. ومن ثم نستدعي الداله بشكل عادي

## التنفيذ



- في البدايه وللتبسيط وقبل أن نرى موضوع RMI Deployment ، ضع كل الملفات في مجلد واحد ، نريد أن نختبر البرنامج في Local Machine .
- الآن قم أولاً بترجمه جميع الملفات:  
`javac *.java`  
الآن ستخرج لك ٤ ملفات `.class`.
- قم بترجمه ملف `HelloImpl.class` باستخدام المترجم الخاص بـ: `rmi`  
`rmic HelloImpl`  
(من غير كتابه الأمتداد.. )

## التنفيذ



- الناتج هنا سوف يكون ملف Stub باسم . HelloImpl\_Stub.class هذا الملف يجب أن يتواجد عند الكلاينت حتى يعمل البرنامج بشكل صحيح هو والملف Hello.class لأننا في الكلاينت سوف نقوم بعمل Cast لهذا النوع) .. لذلك عليك أن نتسخ هذين الملفات الى الكلاينت في حال كان مجلد الكلاينت في مكان آخر أو في جهاز ثاني .. حاليا انسى الكلام واستمر على أساس أنهم جميعهم في نفس المسار ، وبعد قليل نتناول هذا الموضوع بشكل أوسع.
- الآن (في جهه السيرفر ، لكننا حاليا نعمل وجميعهم في جهاز واحد ) قم بتشغيل ملف ..  
rmiregistry وذلك من خلال سطر الأوامر أكتب:  
rmiregistry
- ولن يكون هناك مؤشر أو جملة طباعه تدل على أنه يعمل .. فقط ستجد أن title Bar لسطر الأوامر أصبح يحتوي على rmiregistry وهو دليل على عمل هذه الخدمه..
- : الآن شغل الخادم في نافذه اوامر جديده  
java server

## التنفيذ



- وستجد أن الجملة..... Binding Complete تم طباعتها على الشاشة ..وهكذا يكون السيرفر يعمل ، والخدمه registry تعمل ايضا .. لن تستطيع ايقافها الا بالضغط على CTRL+C
- أخيرا شغل ملف الكلاينت :  
java Client
- وستجد جملة Hello Distributed Computing امامك في الشاشة (حيث تم استدعاء الداله getHelloMessage في السيرفر ، والنتاج من هذه الداله هذا ال String الذي تم طباعته في الكلاينت .. )
- تأكد من فهم ماذا حصل ،، تم استدعاء الداله getHelloMessage الموجوده في السيرفر ، وانتفذت هناك والقيمه الراجعه رجعت للعميل ، الذي قام بطباعتها على الشاشة .. هذه هي RMI بمنتهى البساطه.