**Part 2 - Creating a remoting service and a web service**

In questo video parleremo della creazione di un semplice remoting e di un servizio web. Questa è la continuazione della Parte 1. Guarda la Parte 1 del video tutorial WCF prima di procedere.

Abbiamo 2 client e dobbiamo implementare un servizio per loro. 1. Il primo client utilizza un'applicazione Java per interagire con il nostro servizio, quindi per l'interoperabilità questo client desidera che i messaggi siano in formato XML e che il protocollo sia HTTP.

2. Il secondo client utilizza .NET, quindi per prestazioni migliori desidera che i messaggi siano formattati in binario tramite protocollo TCP.

Per soddisfare il requisito del primo client, creiamo un servizio Web. I servizi Web utilizzano il protocollo HTTP e il formato di messaggio XML. Quindi, il requisito di interoperabilità del primo client sarà soddisfatto. I servizi Web possono essere utilizzati da qualsiasi client creato su qualsiasi piattaforma.

**To create the web service**  
**1.** Create an **empty asp.net web application** project and name it **WebServicesDemo.**  
2. Right click on the project name in solution explorer and add a web service. Name it **HelloWebService**.  
3. Copy and paste the following code **HelloWebService.asmx.cs.**

using System.Web.Services;  
namespace WebServicesDemo  
{  
    [WebService(Namespace = "http://pragimtech.com/WebServices")]  
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]  
    [System.ComponentModel.ToolboxItem(false)]  
    [System.Web.Script.Services.ScriptService]  
    public class HelloWebService : System.Web.Services.WebService  
    {  
        [WebMethod]  
        public string GetMessage(string name)  
        {  
            return "Hello " + name;  
        }  
    }  
}  
  
**Build the solution.**

**Creating a client for the Web Service.**  
**1.** Right click on **WebServicesDemo** solution in solution explorer and add a new asp.net empty web application project and name it **HelloWebApplication**.  
2. Right click on **References** folder in **HelloWebApplication** project and select **Add Service Reference** option. In **Add Service Reference** dialog box click the **Discover** button. In the **namespace** textbox type **HelloWebService** and click **OK.** This should generate a proxy class to invoke the HelloWebService.

The ASP.NET web application is now able to communicate with the web service. Not just asp.net, a JAVA application can also consume the web service.  
  
**To satisfy the requirement of the second client let's create a .NET remoting service.**  
**Creating a remoting service**  
**1.** Create a new **Class Library project** and name it **IHelloRemotingService.**  
**2.** Rename **Class1.cs** file to **IHelloRemotingService.cs.** Copy and paste the following code in

**IHelloRemotingService.cs** file.  
namespace IHelloRemotingService  
{  
    public interface IHelloRemtingService  
    {  
        string GetMessage(string name);  
    }  
}

**3.** Right click on **IHelloRemotingService** solution in solution explorer and add new class library project, and name it **HelloRemotingService**.  
**4.** We want to use interface **IHelloRemotingService** in **HelloRemotingService** project. So add a reference to **IHelloRemotingService** project.

5. Rename **Class1.cs** file to **HelloRemotingService.cs.** Copy and paste the following code in **HelloRemotingService.cs** file.  
using System;  
namespace HelloRemotingService  
{  
    public class HelloRemotingService : MarshalByRefObject,  
        IHelloRemotingService.IHelloRemtingService  
    {  
        public string GetMessage(string name)  
        {  
            return "Hello " + name;  
        }  
    }  
}

**6.** Now we need to **host the remoting service.** To host it let's use a **console application.** A windows application or IIS can also be used to host the remoting service. Right click on **IHelloRemotingService** solution in solution explorer and add **new Console Application  project,** and name it **RemotingServiceHost.**  
**7.** Add a reference to **IHelloRemotingService** and **HelloRemotingService** projects and **System.Runtime.Remoting** assembly.  
**8.** Copy and paste the following code in **Program.cs** file

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Runtime.Remoting;

using System.Runtime.Remoting.Channels;

using System.Runtime.Remoting.Channels.Tcp;

namespace RemotingServiceHost

{

class Program

{

static void Main()

{

HelloRemotingService.HelloRemotingService remotingService =

new HelloRemotingService.HelloRemotingService();

TcpChannel channel = new TcpChannel(8080);

ChannelServices.RegisterChannel(channel);

RemotingConfiguration.RegisterWellKnownServiceType(

typeof(HelloRemotingService.HelloRemotingService), "GetMessage",

WellKnownObjectMode.Singleton);

Console.WriteLine("Host started @ " + DateTime.Now.ToString());

Console.ReadLine();

}

}

}

**9.** Now we need to create the **client for our remoting service.** Let's use **windows application as the client.** Right click on **IHelloRemotingService** solution in solution explorer and **add new windows application.**  
**10.** Add a reference to **IHelloRemotingService** project and **System.Runtime.Remoting** assembly.

**7.** Add a reference to **IHelloRemotingService** and **HelloRemotingService** projects and **System.Runtime.Remoting** assembly.  
**8.** Copy and paste the following code in **Program.cs** file  
using System;  
using System.Runtime.Remoting;  
using System.Runtime.Remoting.Channels;  
using System.Runtime.Remoting.Channels.Tcp;  
  
namespace RemotingServiceHost  
{  
    class Program  
    {  
        static void Main()  
        {  
            HelloRemotingService.HelloRemotingService remotingService =  
                new HelloRemotingService.HelloRemotingService();  
            TcpChannel channel = new TcpChannel(8080);  
            ChannelServices.RegisterChannel(channel);  
            RemotingConfiguration.RegisterWellKnownServiceType(  
               typeof(HelloRemotingService.HelloRemotingService), "GetMessage",  
               WellKnownObjectMode.Singleton);  
            Console.WriteLine("Host started @ " + DateTime.Now.ToString());  
            Console.ReadLine();  
        }  
    }  
}

**9.** Now we need to create the **client for our remoting service.** Let's use **windows application as the client.** Right click on **IHelloRemotingService** solution in solution explorer and **add new windows application.**

**10. Add a reference to IHelloRemotingService project and System.Runtime.Remoting assembly.**

**11. Drag and drop a textbox, button and a label control on Form1 in the windows application.  
12. Double click the button to generate the click event handler. Copy and paste the following code in Form1.cs.  
using System;  
using System.Runtime.Remoting.Channels;  
using System.Runtime.Remoting.Channels.Tcp;  
using System.Windows.Forms;  
  
namespace HelloRemotingServiceClient  
{  
    public partial class Form1 : Form  
    {  
        IHelloRemotingService.IHelloRemtingService client;  
        public Form1()  
        {  
            InitializeComponent();  
            TcpChannel channel = new TcpChannel();  
            ChannelServices.RegisterChannel(channel);  
            client = (IHelloRemotingService.IHelloRemtingService)Activator.GetObject(  
              typeof(IHelloRemotingService.IHelloRemtingService),   
              "tcp://localhost:8080/GetMessage");  
        }  
  
        private void button1\_Click(object sender, EventArgs e)  
        {  
            label1.Text = client.GetMessage(textBox1.Text);  
        }  
    }  
}**