**Part 3 - Creating a wcf service**

Questa è la continuazione alla parte 2. Si prega di guardare la parte 2 dal tutorial video WCF prima di procedere.

In this video, we will discuss   
**1.** Creating a WCF service  
**2.** Hosting the WCF service using a console application  
**3.** Exposing 2 service endpoints.   
**4.** Creating a windows and a web Client applications.

**Let's take the scenario that we discussed in**[**Part 2**](http://csharp-video-tutorials.blogspot.com/2013/11/part-2-creating-remoting-service-and_17.html)**.**

Prendiamo lo scenario di cui abbiamo discusso nella parte 2.

Abbiamo 2 clienti e dobbiamo implementare un servizio A per loro.

1. Il primo client sta utilizzando un'applicazione Java per interagire con il nostro servizio, quindi per l'interoperabilità questo client vuole che Meesages sia in formato XML e il protocollo sia HTTP.

2. Il secondo client utilizza .NET, quindi per prestazioni migliori questo client desidera formarsi in protocollo binario tramite TCP.

Nella parte 2,

Per soddisfare i requisiti del primo cliente, abbiamo implementato un servizio Web e per soddisfare i requisiti del secondo cliente abbiamo implementato un servizio di ristrutturazione.

In questo video, creeremo un singolo servizio WCF e configureremo 2 endpoint per soddisfare i requisiti di entrambi i client.

**Creating the WCF Service:**  
**1.** Create a new Class Library Project and name it **HelloService.**  
**2.** Delete **Class1.cs**file that is auto-generated.  
**3.** Add a new **WCF Service**with name = **HelloService.**This should automatically generate 2 files (**HelloService.cs** & **IHelloService.cs**). Also a reference to **System.ServiceModel**assembly is added.  
**4.** Copy and paste the following code in **IHelloService.cs** file

using System.ServiceModel;  
namespace HelloService  
{  
    [ServiceContract(Namespace="http://PragimTech.com/ServiceVersion1")]  
    public interface IHelloService  
    {  
        [OperationContract]  
        string GetMessage(string name);  
    }  
}  
**5.** Copy and paste the following code in **HelloService.cs**  
namespace HelloService  
{  
    public class HelloService : IHelloService  
    {  
        public string GetMessage(string name)  
        {  
            return "Hello " + name;  
        }  
    }  
}

Ecco fatto che abbiamo finito di implementare un servizio WCF. Il prossimo passo è ospitare il servizio utilizzando un'applicazione console. Un servizio WCF può anche essere ospitato in un'applicazione Windows o Servizio Windows o IIS. Discuteremo queste opzioni di hosting in una successiva sessione video.

Ospitare il servizio WCF utilizzando un'applicazione console.

**1.** Right click on **HelloService**solution in Solution Explorer and add a new Console Application project with name = **HelloServiceHost**  
**2.** Add a reference to **System.ServiceModel**assembly and **HelloService**project  
**3.** Right click on **HelloServiceHost**project and add **Application Configuration File.**This should add **App.config**file to the project. Copy and paste the following XML. Notice that we have specified **2 endpoints**in the configuration. One endpoint uses **basicHttpBinding,**which communicates over **HTTP**protocol using **XML**messages. This endpoint will satisfy the requirement of the first client. The other endpoint uses **netTcpBinding,**which communicates over **TCP**protocol using **binary messages.**This endpoint will satisfy the requirement of the second client.