

## Bài thực hành 04

### Tạo ứng dụng WCF đầu tiên

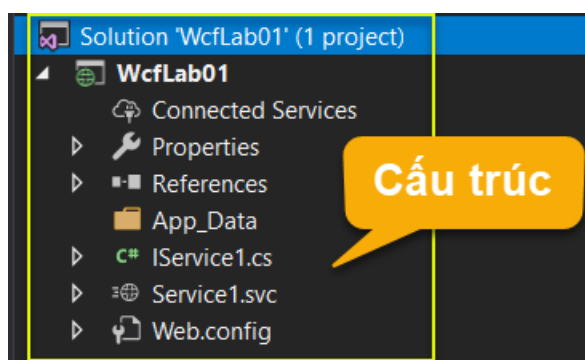
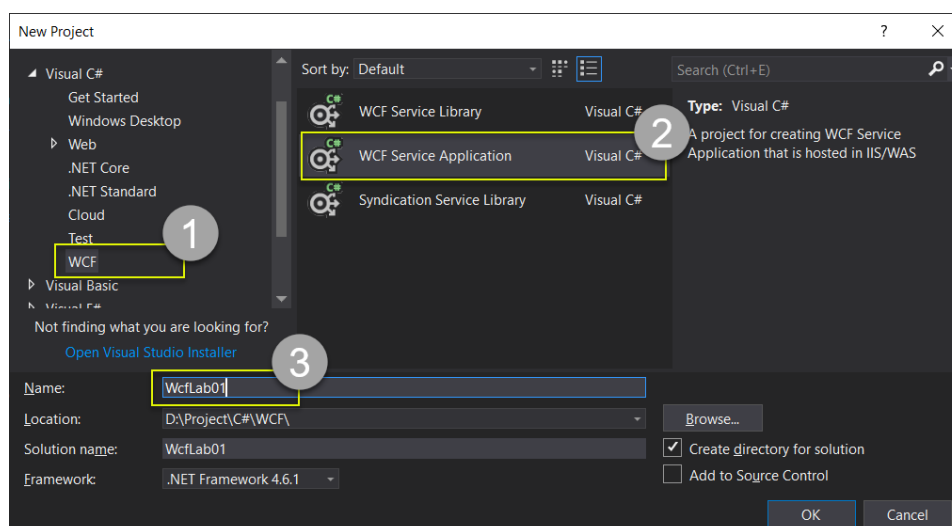
#### MỤC TIÊU

- Tạo WCF Services
- Cấu hình và host WCF Services
- Sử dụng WCF Services

#### BÀI 1: TẠO ỨNG DỤNG WCF SERVICE TRẢ VỀ TÊN VÀ TUỔI CỦA MỘT NGƯỜI NÀO ĐÓ

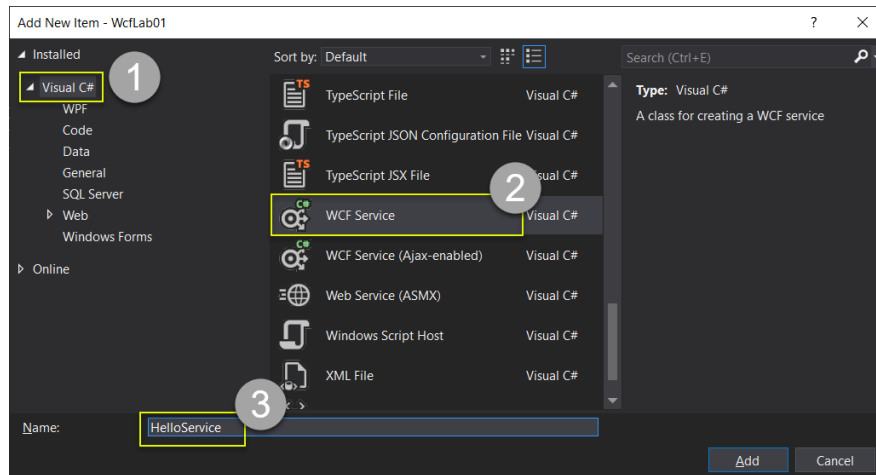
**Bước 1:** Sinh viên tạo một dự án WCF mới đặt tên là **WcfLab01**

Mở Visual Studio 2017 lên và chọn vào **File -> New -> Project**, chọn **WCF Service**

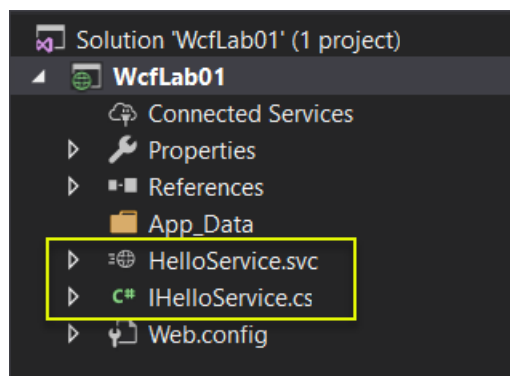


**Bước 2:** Xóa 2 file IService.cs Service.svc

Tạo một service mới bằng cách click chuột phải lên thư mục gốc Add -> New Item, tìm và chọn Wcf Service, Đặt tên là HelloService



Cấu trúc file thư mục có dạng



**Bước 3:** Mở file **IHelloService.cs** khai báo như sau

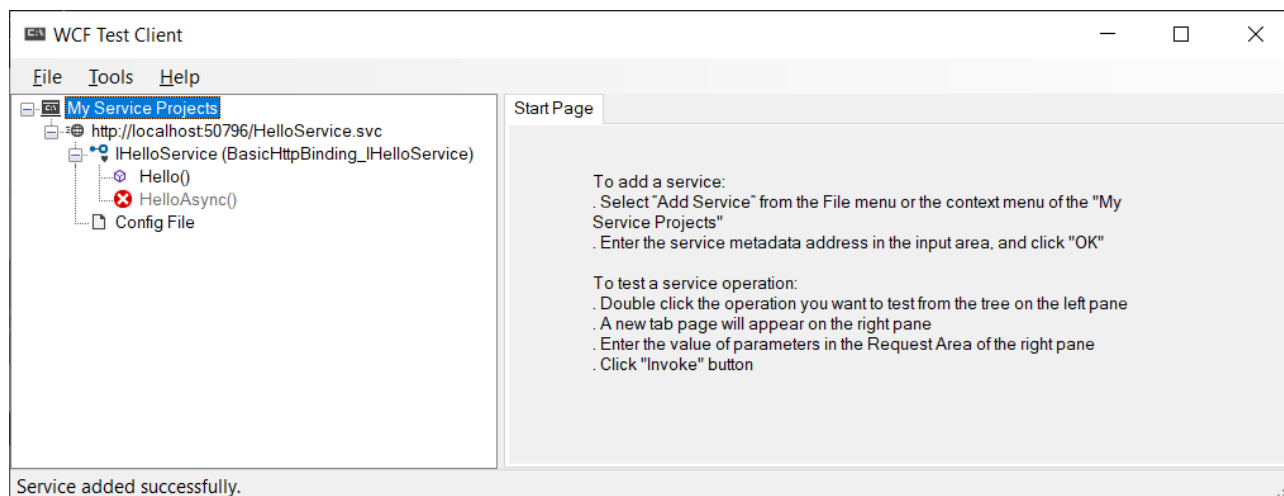
```
using System.ServiceModel;
namespace WcfLab01
{
    [ServiceContract]
    public interface IHelloService
    {
        [OperationContract]
        string Hello(string name, int age);
    }
}
```

**Bước 4:** Mở file **HelloService.svc** và code triển khai phương thức Hello như sau

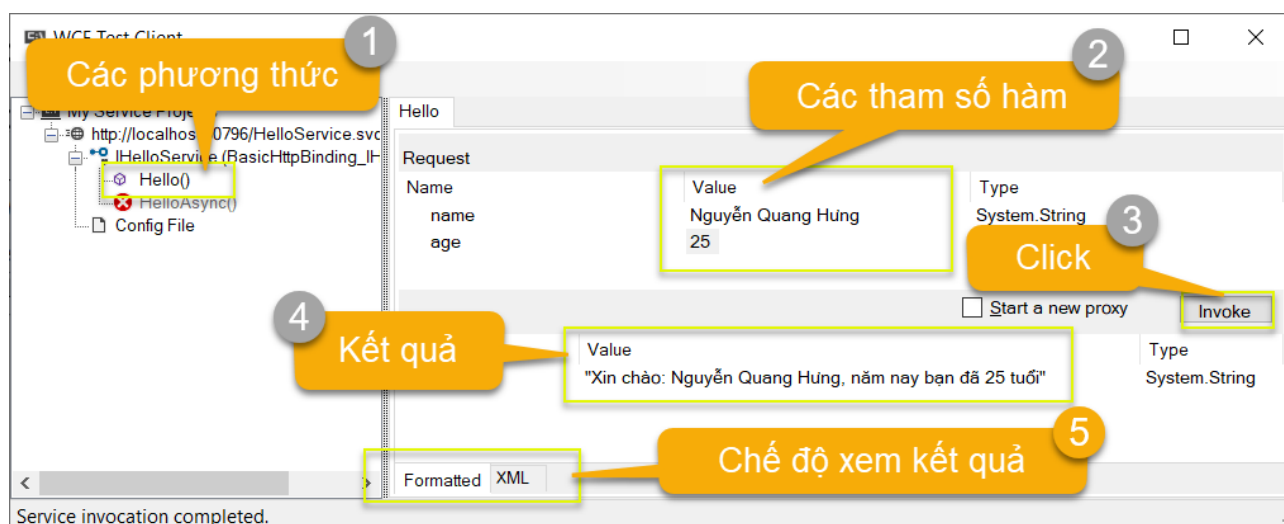
```
using System;
namespace WcfLab01
{
    public class HelloService : IHelloService
    {
        public string Hello(string name, int age)
        {
            return String.Format("Xin chào: {0}, năm nay bạn đã {1} tuổi", name, age);
        }
    }
}
```

**Bước 5:** Build lại dự án và thực hiện một trong 2 cách sau để chạy test dự án

**Cách 1:** Nhấn CTRL + F5 dự án sẽ được khởi chạy và ứng dụng WCF test client cũng được khởi động có dạng



Click chọn hàm **Hello()** và nhập các tham số thích hợp sau đó click vào **Invoke** như hình sau



**Cách 2:** Click phải chuột trực tiếp lên file **HelloService.svc** và chọn **View In Browser**

Trình duyệt sẽ mở trực tiếp file **HelloService.svc** trên link có dạng

**http://localhost:50796/HelloService.svc** có màn hình tương tự

## HelloService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://localhost:50796/HelloService.svc?wsdl
```

You can also access the service description as a single file:

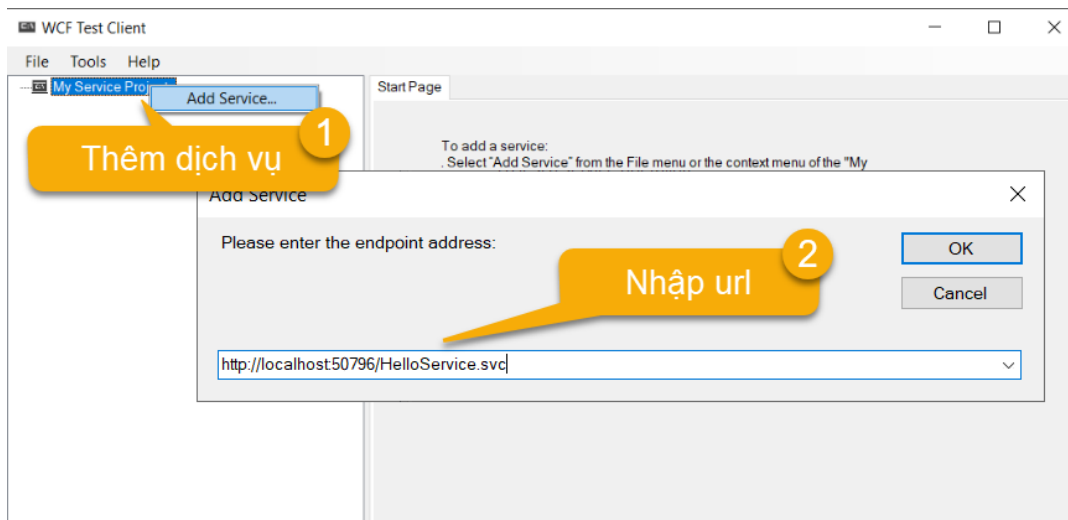
```
http://localhost:50796/HelloService.svc?singleWsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

Với cách này, muốn test trên ứng dụng WCF test client thì cần mở trực tiếp ứng dụng WCF test Client tại thư mục

**C:\Users\User\_name\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\WcfTestClient.exe**

Sau đó click phải chuột lên **My Service Project** chọn **Add Service** và nhập url có dạng như trên vào ô như hình dưới đây

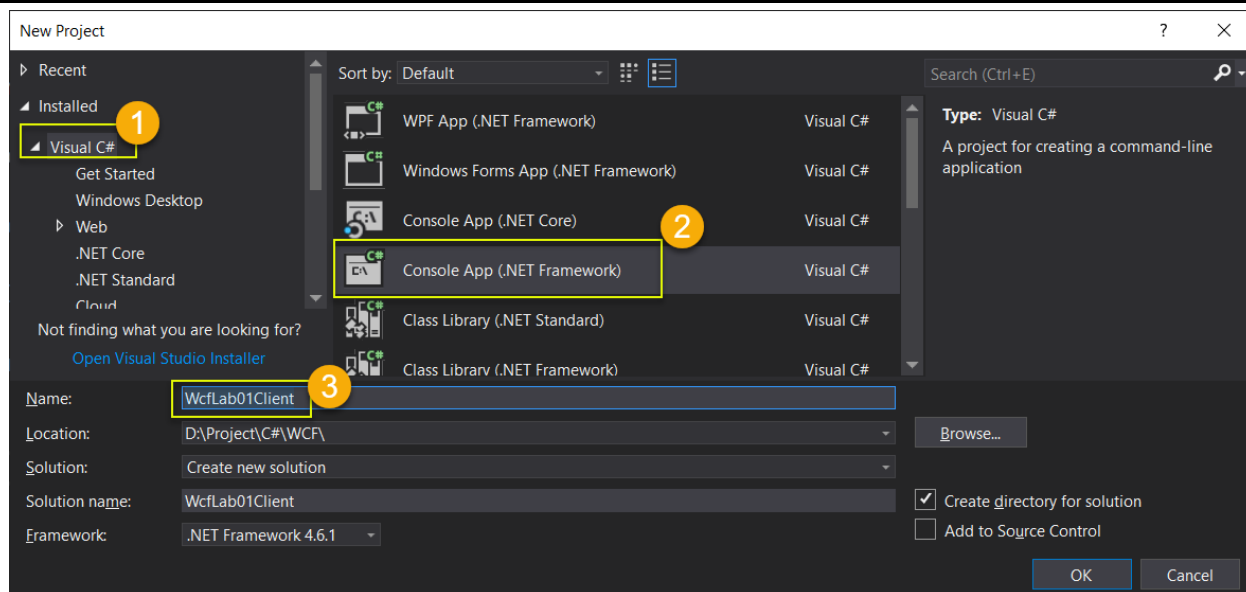


Sau đó tiến hành test như cách 1

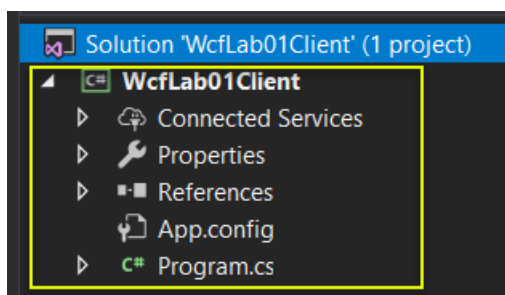
**Bước 6:** Tạo một dự án Console App để test Service đã tạo ở trên

**Chú ý:** Service vẫn đảm bảo đang được khởi chạy

Mở Visual Studio 2017 lên trong phiên mới và tạo project mới đặt tên cho project là **WcfLab01Client** chọn như hình

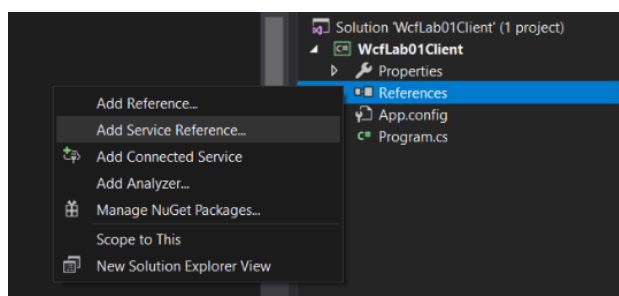


Cấu trúc file thư mục:

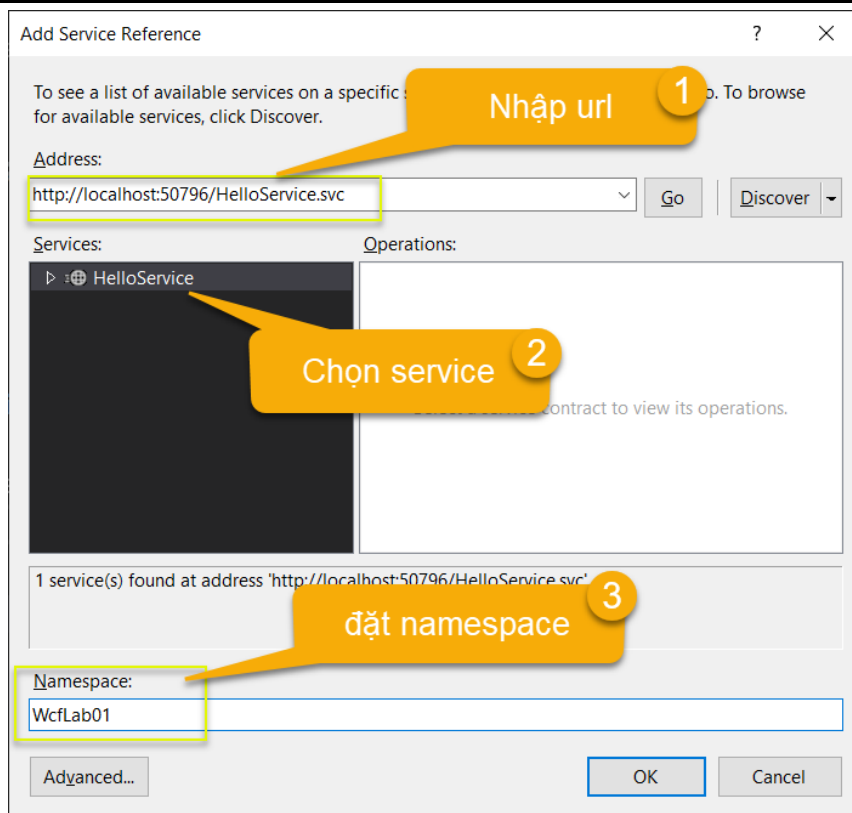


**Bước 7:** Thêm dịch vụ đã tạo ở trên vào dự án này bằng cách

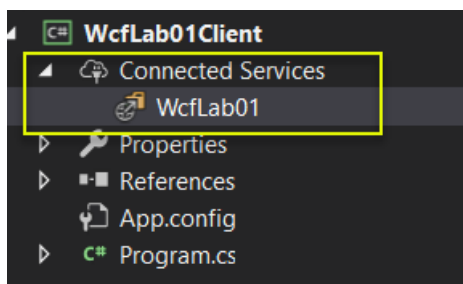
Click phải chuột lên thư mục References chọn Add Service Reference..



Sau đó nhập url của WCF Service đã tạo ở trên như hình sau

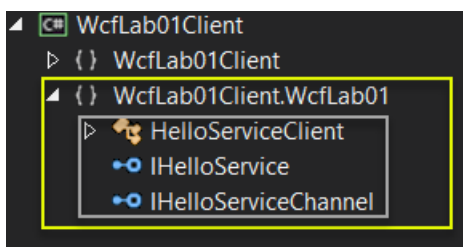


Kết quả dịch vụ được add vào như sau



Nếu click vào WcfLab01 ở trên có thể thấy các thành phần của HelloService tại đây

**Chú ý:** HelloServiceClient



**Bước 8:** Mở file Program.cs của dự án **WcfLab01Client** này và sử dụng HelloServiceClient như sau

```
using System;
using System.Text;
using WcfLab01Client.MyService;

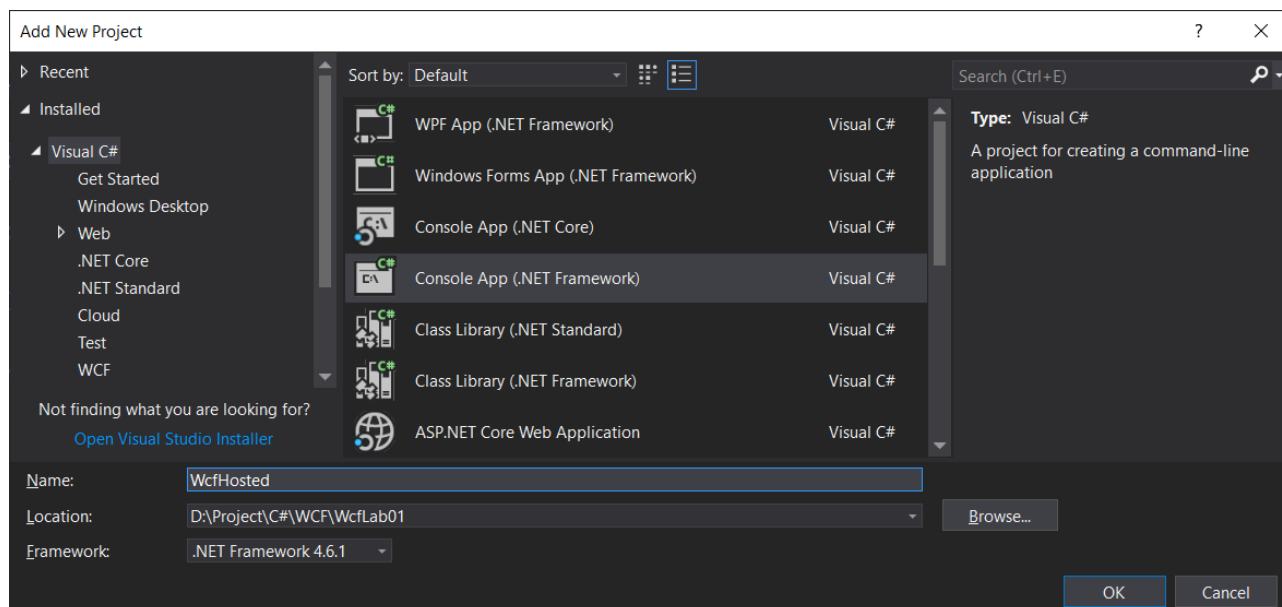
namespace WcfLab01Client
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.OutputEncoding = Encoding.UTF8;
            HelloServiceClient client = new HelloServiceClient();
            string hello = client.Hello("NGUYEN HUY HOANG", 25);
            Console.WriteLine(hello);
            Console.ReadLine();
        }
    }
}
```

Build lại dự án **WcfLab01Client** và chạy sau đó xem kết quả có dạng

```
❯ Select C:\Windows\system32\cmd.exe
Xin chào: NGUYEN HUY HOANG, năm nay bạn đã 25 tuổi
```

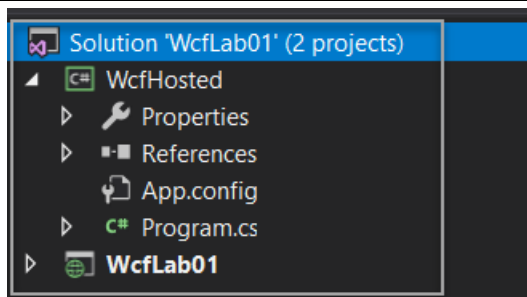
## BÀI 2: TẠO HOST VÀ CUSTOM ENDPOINT

**Bước 1:** Mở lại dự án WcfLab01, click lên Solution và add Console App, đặt tên là WcfHosted

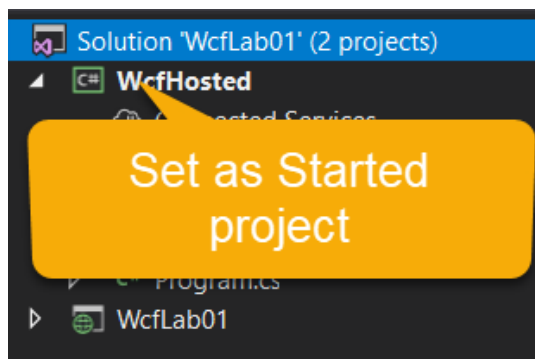


Cấu trúc Solution hiện tại như sau

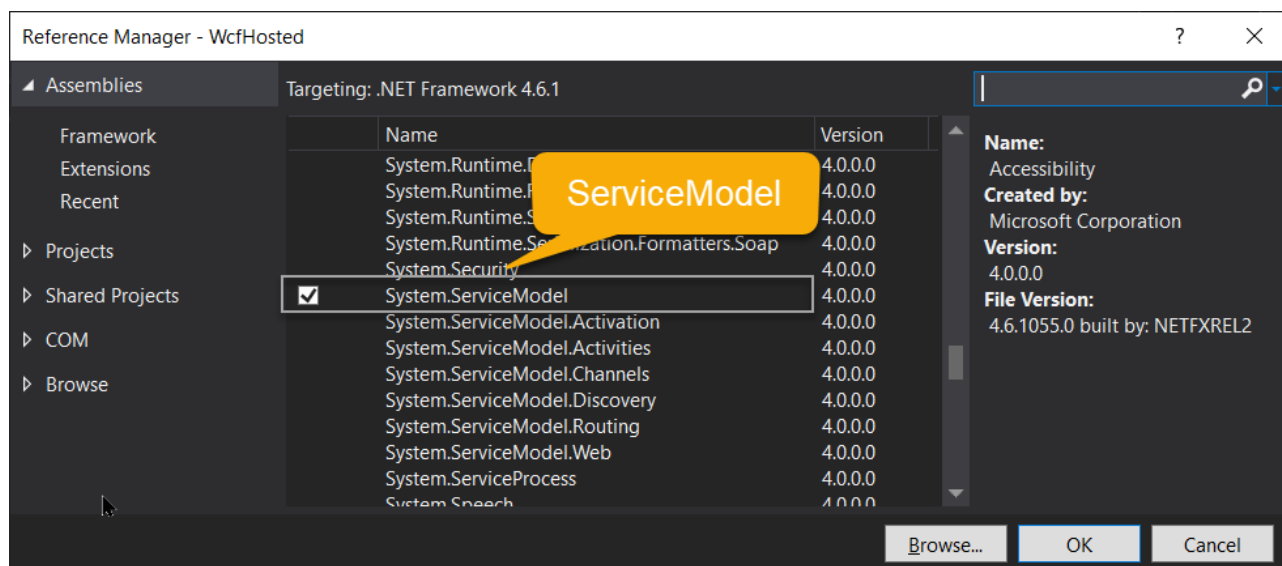




Chú ý: Click phải chuột lên WcfHosted project và chọn **Set as StartUp Project** để chọn làm dự án chạy chính khi build

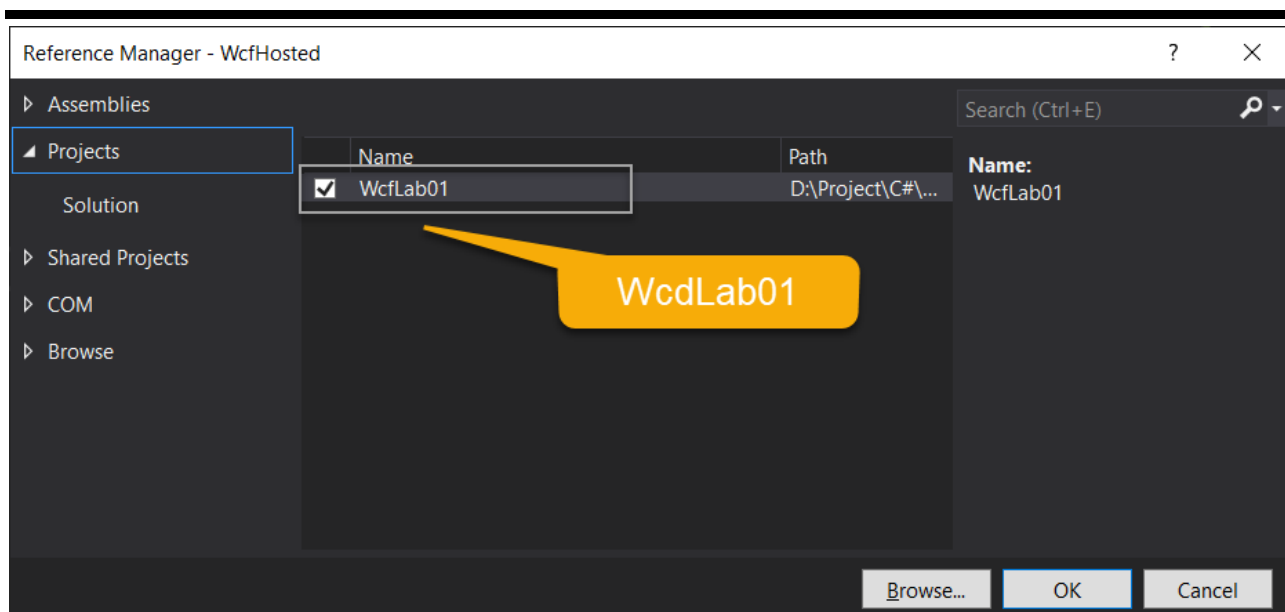


**Bước 2:** Add thư viện ServiceModel bằng cách click phải chuột lên thư mục References và chọn Add Reference..., tìm đến **System.ServiceModel** như hình sau



**Bước 3:** Add thêm service của WcfLab01 bằng cách click phải chuột vào thư mục References chọn Add Reference... chọn đến Project và chọn WcfLab01 như hình sau





**Bước 4:** Khai báo endpoint, Mở file App.config của dự án **WcfHosted** lên và và thêm code cấu hình cho endpoint

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
  <system.serviceModel>
    <!--Enabled - publish metadata-->
    <behaviors>
      <serviceBehaviors>
        <behavior name="WcfLab01">
          <serviceMetadata httpGetEnabled="true"/>
        </behavior>
      </serviceBehaviors>
    </behaviors>
    <!--Cấu hình các endpoint-->
    <services>
      <service name="WcfLab01.HelloService" behaviorConfiguration="WcfLab01">
        <host>
          <baseAddresses>
            <add baseAddress="http://localhost:8000" />
          </baseAddresses>
        </host>
        <endpoint address="HelloService" binding="basicHttpBinding" contract="WcfLab01.IHelloService" />
      </service>
    </services>
  </system.serviceModel>
</configuration>
```

**Bước 5:** Mở file Program.cs của dự án WcfHosted này và code mở cổng dịch vụ như sau

```
using System;
using System.ServiceModel;

namespace WcfHosted
{
  class Program
  {
    static void Main(string[] args)
    {
      // Khởi tạo ServiceHost
      ServiceHost host = new ServiceHost(typeof(WcfLab01.HelloService));
```

```
host.Open(); // mở host
// thông báo
Console.WriteLine("Hosting started on http://localhost:8000");
Console.ReadLine();
}
}
```

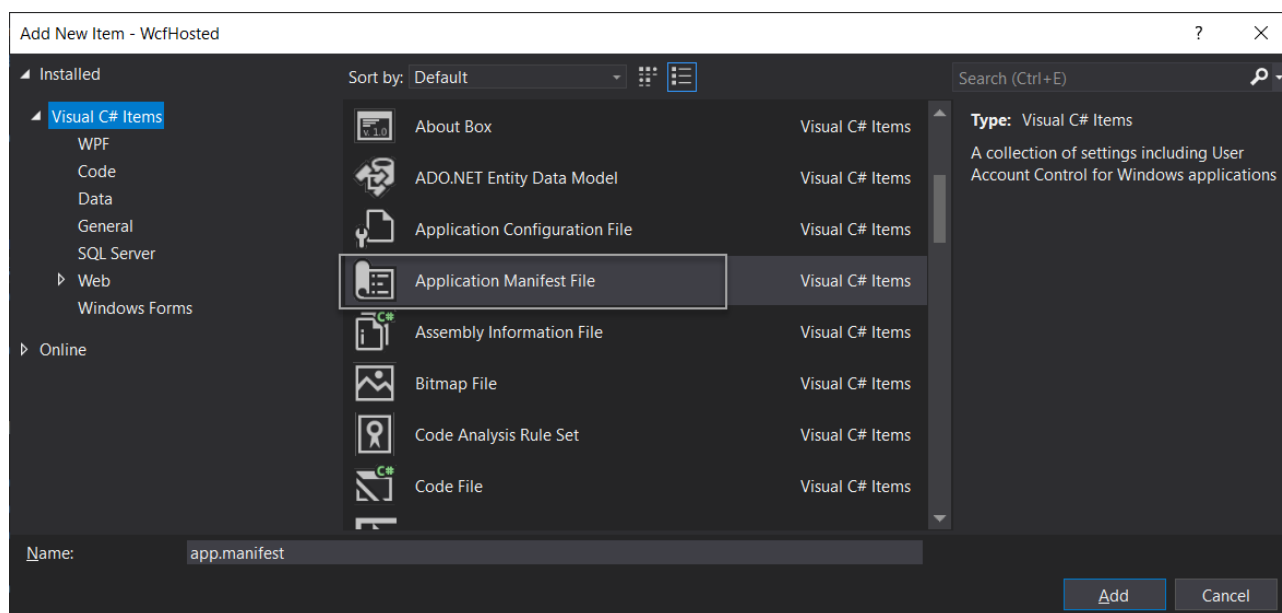
Build lại dự án và khởi chạy sẽ có lỗi tương tự như sau

```
Unhandled Exception: System.ServiceModel.AddressAccessDeniedException: HTTP could not register URL http://+:8000/. Your process does not have access rights to this namespace (see http://go.microsoft.com/fwlink/?LinkId=70353 for details). --> System.Net.HttpListenerException: Access is denied
at System.Net.HttpListener.AddAllPrefixes()
at System.Net.HttpListener.Start()
at System.ServiceModel.Channels.SharedHttpTransportManager.OnOpen()
--- End of inner exception stack trace ---
at System.ServiceModel.Channels.SharedHttpTransportManager.OnOpen()
at System.ServiceModel.Channels.TransportManager.Open(TransportChannelListener channelListener)
at System.ServiceModel.Channels.TransportManagerContainer.Open(SelectTransportManagersCallback selectTransportManagerCallback)
at System.ServiceModel.Channels.TransportChannelListener.OnOpen(TimeSpan timeout)
at System.ServiceModel.Channels.HttpChannelListener`1.OnOpen(TimeSpan timeout)
at System.ServiceModel.Channels.CommunicationObject.Open(TimeSpan timeout)
at System.ServiceModel.Dispatcher.ChannelDispatcher.OnOpen(TimeSpan timeout)
at System.ServiceModel.Channels.CommunicationObject.Open(TimeSpan timeout)
at System.ServiceModel.ServiceHostBase.OnOpen(TimeSpan timeout)
at System.ServiceModel.Channels.CommunicationObject.Open(TimeSpan timeout)
at System.ServiceModel.Channels.CommunicationObject.Open()
at WcfHosted.Program.Main(String[] args) in D:\Project\C#\WCF\WcfLab01\WcfHosted\Program.cs:line 11
Press any key to continue . . .
```

**Bước 6:** Giải quyết lỗi trên, có 2 cách

**Cách 1:** Mở lại Visual Studio 2017 với quyền Administrator rồi build dự án như bình thường

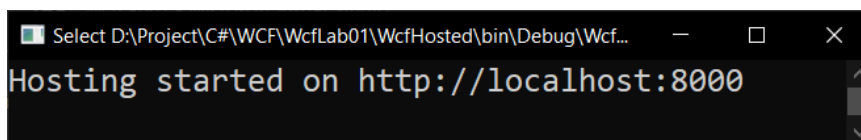
**Cách 2:** Click phải chuột lên dự án **WcfHosted** và chọn Add -> new Item và chọn đến **Application Manifest File**, để nguyên tên file là app.manifest



Sau đó tìm và cấu hình thuộc tính **requestedExecutionLevel** như sau

```
<requestedExecutionLevel level="requireAdministrator" uiAccess="false" />
```

Build lại và CTRL + F5 để chạy dự án hệ thống window sẽ hiển thị cửa sổ UAC để xác nhận, chọn Yes và dự án khởi động bình thường với địa chỉ đã cấu hình



Có thể mở trình duyệt và truy cập: <http://localhost:8000>

Kết quả sẽ tương tự như hình dưới đây

## HelloService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://localhost:8000/?wsdl
```

You can also access the service description as a single file:

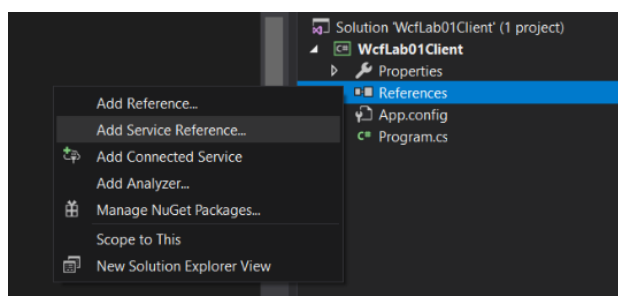
```
http://localhost:8000/?singleWsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

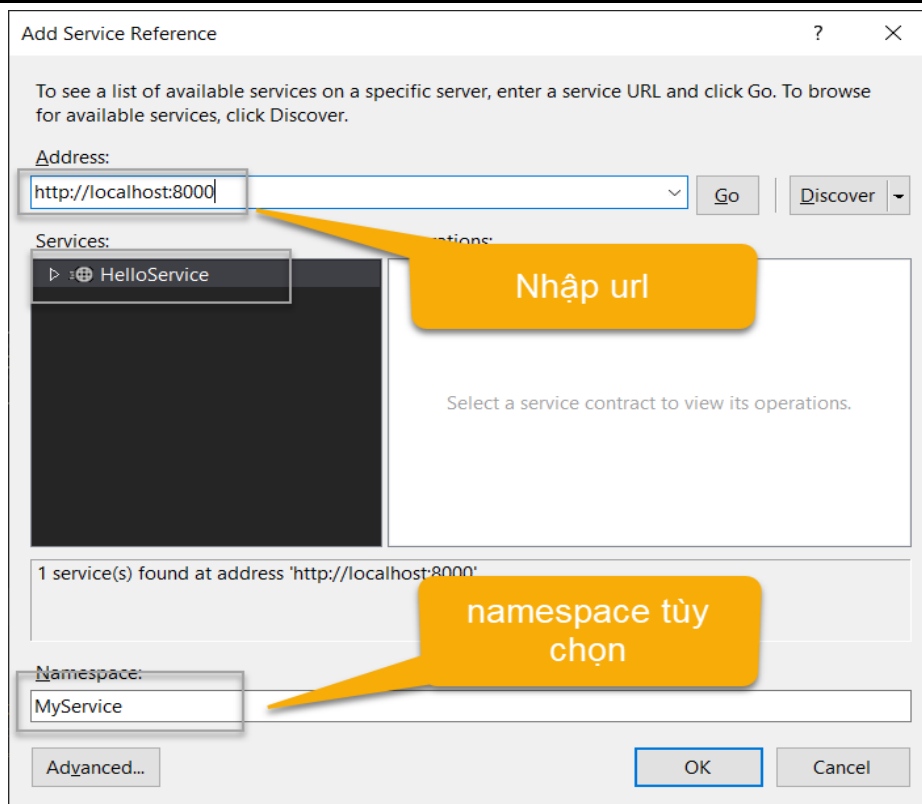
**Bước 7:** Add dịch vụ vừa cấu hình host vào dự án client bằng cách

Mở dự án client **WcfLab01Client** lên (đã tạo ở bài 1)

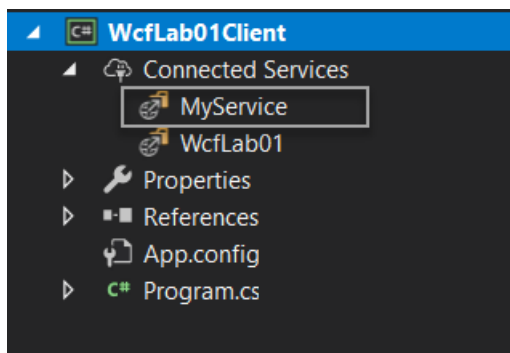
Click phải chuột lên thư mục References chọn Add Service Reference..



Sau đó nhập url <http://localhost:8000> của WCF Service đã tạo ở trên như hình sau



Dịch vụ đã add vào dự án có dạng



**Bước 8:** Mở lại file Program.cs và sử dụng MyService như bình thường

Chú ý: đảm bảo host vẫn được khởi động

```
using System;
using WcfLab01Client.MyService;

namespace WcfLab01Client
{
    class Program
    {
        static void Main(string[] args)
        {
            HelloServiceClient client = new HelloServiceClient();
            string hello = client.Hello("NGUYEN HUY HOANG", 25);
            Console.WriteLine(hello);
            Console.ReadLine();
        }
    }
}
```

Kết quả khi chạy dự án phía Client vẫn như trên

```
Select C:\Windows\system32\cmd.exe
Xin chào: NGUYEN HUY HOANG, năm nay bạn đã 25 tuổi
```

## BÀI TẬP TỰ LÀM

Sinh viên tạo một ứng dụng WCF Service đặt tên tùy ý

Trong ứng dụng tạo AccountService có phương thức checkLogin cần 2 tham số là email và password, phương thức trả về true hoặc false dựa vào 2 tham số đó so sánh với email: [admin@gmail.com](mailto:admin@gmail.com), password: 123456

Tạo thêm dự án Console App để cấu hình endpoint cho dự án có dạng:  
<http://localhost:3500>

Tiếp theo sinh viên tạo một vài dự án Client để test gồm một vài trong các loại dự án Console App, Window form, Web form. ASP .NET MVC, NET CORE, WPF ...

Để test dịch vụ trên bằng cách tạo form đăng nhập và hiển thị kết quả nếu thành công hay thất bại ngay trên form đó