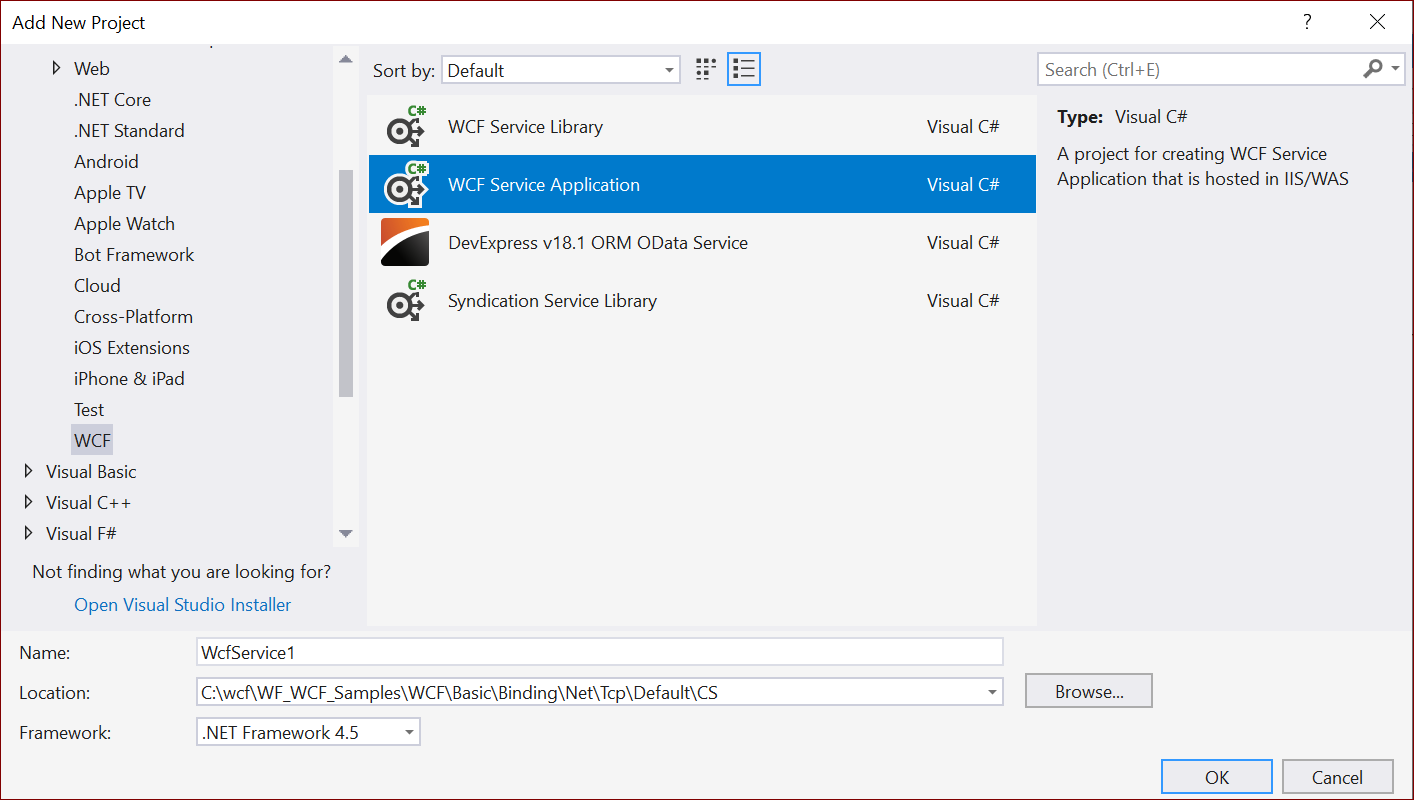
# Hướng dẫn cấu hình MULTI-END POINTS

1. Tạo dự án WCF Service Application



1. Viết code cho dịch vụ

using System.ServiceModel;

namespace Microsoft.ServiceModel.Samples

{

// Define a service contract.

[ServiceContract(Namespace="http://Microsoft.ServiceModel.Samples")]

public interface ICalculator

{

[OperationContract]

double Add(double n1, double n2);

[OperationContract]

double Subtract(double n1, double n2);

[OperationContract]

double Multiply(double n1, double n2);

[OperationContract]

double Divide(double n1, double n2);

}

// Service class which implements the service contract.

public class CalculatorService : ICalculator

{

public double Add(double n1, double n2)

{

return n1 + n2;

}

public double Subtract(double n1, double n2)

{

return n1 - n2;

}

public double Multiply(double n1, double n2)

{

return n1 \* n2;

}

public double Divide(double n1, double n2)

{

return n1 / n2;

}

}

1. Cấu hình End point thông qua web.config

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<system.serviceModel>

<services>

<service

name="Microsoft.ServiceModel.Samples.CalculatorService">

<!-- this endpoint is exposed at the base address provided by host: http://localhost/servicemodelsamples/service.svc -->

<endpoint address=""

binding="basicHttpBinding"

contract="Microsoft.ServiceModel.Samples.ICalculator" />

<!-- secure endpoint exposed at {base address}/secure: http://localhost/servicemodelsamples/service.svc/secure -->

<endpoint address="secure"

binding="wsHttpBinding"

contract="Microsoft.ServiceModel.Samples.ICalculator" />

<!-- the mex endpoint is exposed at http://localhost/servicemodelsamples/service.svc/mex -->

<endpoint address="mex"

binding="mexHttpBinding"

contract="IMetadataExchange" />

</service>

</services>

<!--For debugging purposes set the includeExceptionDetailInFaults attribute to true-->

<behaviors>

<serviceBehaviors>

<behavior>

<serviceMetadata httpGetEnabled="True"/>

<serviceDebug includeExceptionDetailInFaults="False" />

</behavior>

</serviceBehaviors>

</behaviors>

</system.serviceModel>

</configuration>

1. Tạo Proxy và config file để sử dụng cho client (bằng công cụ svcutil hoặc add service references)

App.config của client như sau:

<?xml version="1.0"?>

<configuration>

<system.serviceModel>

<client>

<!-- passing "basic" into the constructor of the CalculatorClient class selects this endpoint-->

<endpoint name="basic" address="http://localhost/servicemodelsamples/service.svc" binding="basicHttpBinding" contract="Microsoft.ServiceModel.Samples.ICalculator"/>

<!-- passing "secure" into the constructor of the CalculatorClient class selects this endpoint-->

<endpoint name="secure" address="http://localhost/servicemodelsamples/service.svc/secure" binding="wsHttpBinding" contract="Microsoft.ServiceModel.Samples.ICalculator"/>

</client>

</system.serviceModel>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>

</startup>

</configuration>

1. Mã nguồn của Client

class Client

{

static void Main()

{

// Create a client to the basic endpoint configuration

CalculatorClient client = new CalculatorClient("basic");

Console.WriteLine("Communicate with basic endpoint.");

// call operations

DoCalculations(client);

// close the client and release resources

client.Close();

// Create a client to the secure endpoint configuration

client = new CalculatorClient("secure");

Console.WriteLine("Communicate with secure endpoint.");

// call operations

DoCalculations(client);

// close the client and release resources

client.Close();

Console.WriteLine();

Console.WriteLine("Press <ENTER> to terminate client.");

Console.ReadLine();

}

static void DoCalculations(CalculatorClient client)

{

// Call the Add service operation.

double value1 = 100.00D;

double value2 = 15.99D;

double result = client.Add(value1, value2);

Console.WriteLine("Add({0},{1}) = {2}", value1, value2, result);

// Call the Subtract service operation.

value1 = 145.00D;

value2 = 76.54D;

result = client.Subtract(value1, value2);

Console.WriteLine("Subtract({0},{1}) = {2}", value1, value2, result);

// Call the Multiply service operation.

value1 = 9.00D;

value2 = 81.25D;

result = client.Multiply(value1, value2);

Console.WriteLine("Multiply({0},{1}) = {2}", value1, value2, result);

// Call the Divide service operation.

value1 = 22.00D;

value2 = 7.00D;

result = client.Divide(value1, value2);

Console.WriteLine("Divide({0},{1}) = {2}", value1, value2, result);

}

}