**Hướng dẫn tạo WCF với 5 bước**

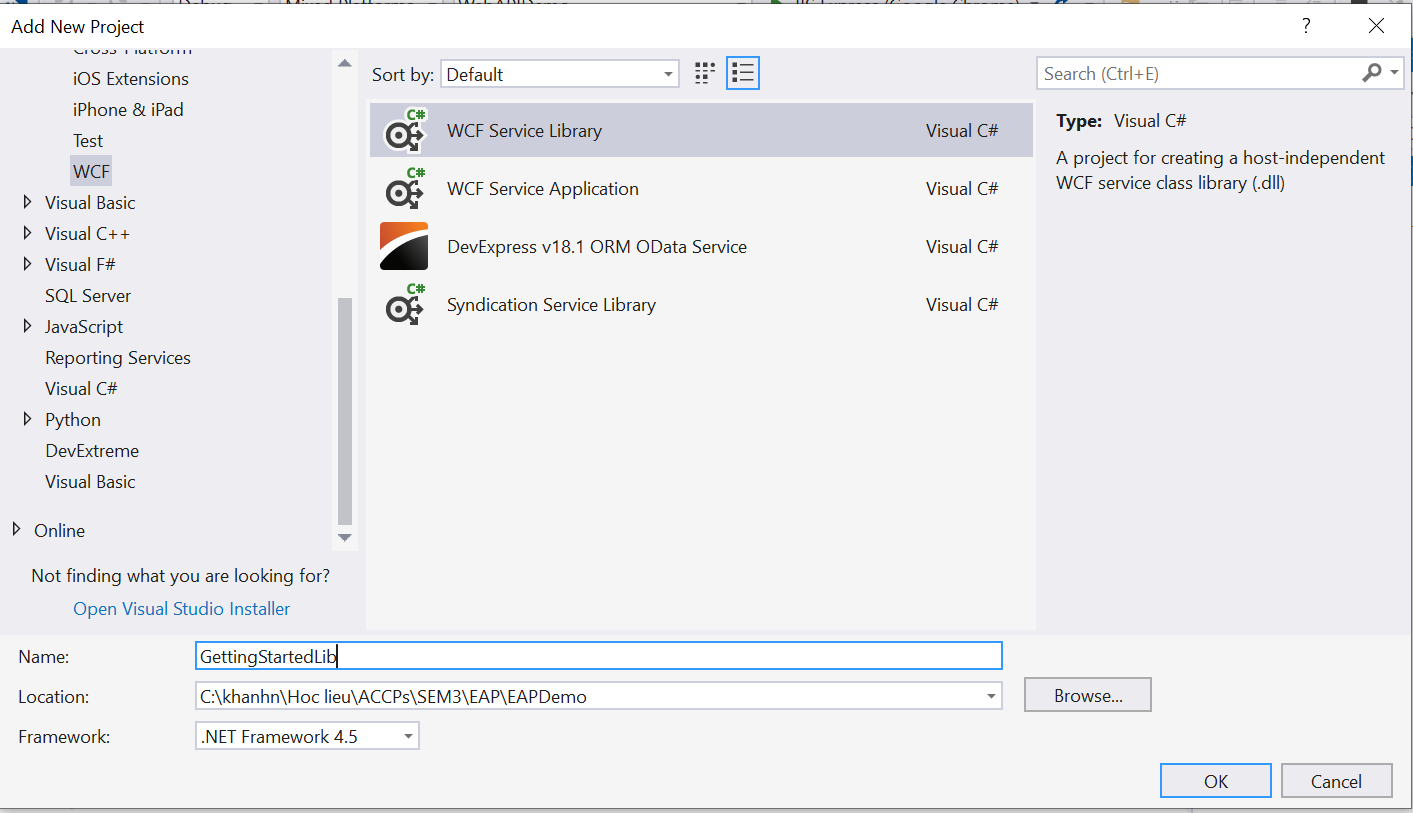
# Tạo service contract

* Implement service contract
* Host và chạy service
* Tạo Client service
* Sử dụng Client service

# Bước 1: Tạo service contract

# Create a WCF Service Library project and define a service contract interface

1. Mở Visual Studio dưới quyền administrator
2. Tạo dự án mới và chọn loại dự án là **WCF Service Library**



1. Đặt tên dự án là *GettingStartedLib*
2. Trong cửa sổ **Solution Explorer**, mở file **IService1.cs** và thay bằng đoạn mã dưới đây:

using System;

using System.ServiceModel;

namespace GettingStartedLib

{

[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);

}

}

# Bước 2: Implement service contract

|  |
| --- |
| Mở file **Service1.cs và sửa mã như sau** |
| public class CalculatorService : ICalculator  {  public double Add(double n1, double n2)  {  double result = n1 + n2;  Console.WriteLine("Received Add({0},{1})", n1, n2);  // Code added to write output to the console window.  Console.WriteLine("Return: {0}", result);  return result;  }  public double Subtract(double n1, double n2)  {  double result = n1 - n2;  Console.WriteLine("Received Subtract({0},{1})", n1, n2);  Console.WriteLine("Return: {0}", result);  return result;  }  public double Multiply(double n1, double n2)  {  double result = n1 \* n2;  Console.WriteLine("Received Multiply({0},{1})", n1, n2);  Console.WriteLine("Return: {0}", result);  return result;  }  public double Divide(double n1, double n2)  {  double result = n1 / n2;  Console.WriteLine("Received Divide({0},{1})", n1, n2);  Console.WriteLine("Return: {0}", result);  return result;  }  } |
| Thay đổi trong file App.config với các dòng highlight ở dòng 11, 12 và 20 như sau: |
|  |

Biên dịch mã nguồn

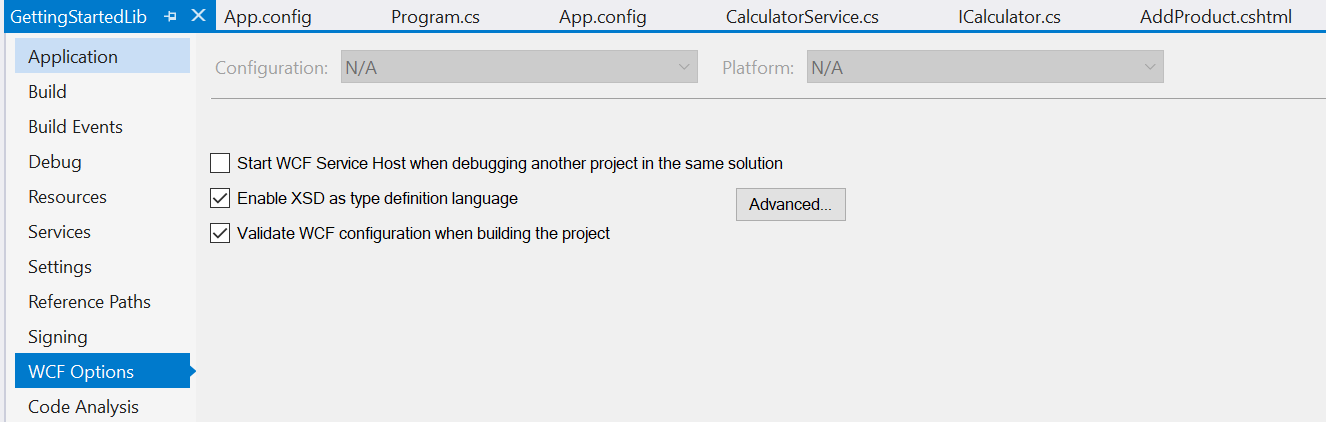
# Bước 3 - Host và chạy service

Tạo dự án Console mới để host wcf, thực hiện theo các bước sau

1. Tạo dự án console app trong Visual Studio đặt tên **GettingStartedHost**
2. Add tham chiếu dự án **GettingStartedLib**
3. Add tham chiếu [System.ServiceModel](https://docs.microsoft.com/en-us/dotnet/api/system.servicemodel) assembly
4. In the **Solution Explorer** window, select the **References** folder under the **GettingStartedHost** project, and then select **Add Reference** from the shortcut menu.
5. In the **Add Reference** window, under **Assemblies** on the left side of the window, select **Framework**.
6. Select **System.ServiceModel**, and then select **OK**.
7. Save the solution by selecting **File** > **Save All**.
8. Viết mã cho ứng dụng

|  |
| --- |
| using System;  using System.ServiceModel;  using System.ServiceModel.Description;  using GettingStartedLib;  namespace GettingStartedHost  {  class Program  {  static void Main(string[] args)  {  // Step 1: Create a URI to serve as the base address.  Uri baseAddress = new Uri("http://localhost:8000/GettingStarted/");  // Step 2: Create a ServiceHost instance.  ServiceHost selfHost = new ServiceHost(typeof(CalculatorService), baseAddress);  try  {  // Step 3: Add a service endpoint.  selfHost.AddServiceEndpoint(typeof(ICalculator), new WSHttpBinding(), "CalculatorService");  // Step 4: Enable metadata exchange.  ServiceMetadataBehavior smb = new ServiceMetadataBehavior();  smb.HttpGetEnabled = true;  selfHost.Description.Behaviors.Add(smb);  // Step 5: Start the service.  selfHost.Open();  Console.WriteLine("The service is ready.");  // Close the ServiceHost to stop the service.  Console.WriteLine("Press <Enter> to terminate the service.");  Console.WriteLine();  Console.ReadLine();  selfHost.Close();  }  catch (CommunicationException ce)  {  Console.WriteLine("An exception occurred: {0}", ce.Message);  selfHost.Abort();  }  }  }  } |

Click chuột phải trên dự án, bỏ chọn mục **Start WCF**



# Bước 4 - Tạo Client service

* Tạo dự án mới, chọn loại dự án **Console App (.NET Framework)**, và đặt tên là GettingStartedClient.
* Tham chiếu System.ServiceModel
* Tham chiếu dịch vụ
  + Trong cửa sổ **Solution Explorer**, Chọn **References** trong **GettingStartedClient** project, sau đó click chọn **Add Service Reference**.
  + Trong cửa sổ **Add Service Reference**, chọn **Discover**.
  + Dịch vụ CalculatorService khởi động và hiển thị trong danh sách hộp **Services**.
  + Chọn **CalculatorService** và click nút mở rộng để hiển thị service contracts, để tên namespace mặc định và bấm **OK**.
  + Visual Studio adds thư mục **Connected Services** vào dự án **GettingStartedClient**.

### **Sử dụng công cụ ServiceModel Metadata Utility tool**

Các bước dưới đây minh họa cách sử dụng svcutil.exe để tạo ra file proxy và app.config

svcutil.exe /language:cs /out:generatedProxy.cs /config:app.config <http://localhost:8000/GettingStarted/CalculatorService>

Sau khi tạo tham chiếu đến service, file app.config được tạo ra trong dự án như sau

|  |
| --- |
| <?xml version="1.0" encoding="utf-8" ?>  <configuration>  <startup>  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />  </startup>  <system.serviceModel>  <bindings>  <basicHttpBinding>  <binding name="BasicHttpBinding\_ICalculator" />  </basicHttpBinding>  </bindings>  <client>  <endpoint address="http://localhost:8000/GettingStarted/CalculatorService"  binding="basicHttpBinding" bindingConfiguration="BasicHttpBinding\_ICalculator"  contract="ServiceReference1.ICalculator" name="BasicHttpBinding\_ICalculator" />  </client>  </system.serviceModel>  </configuration> |

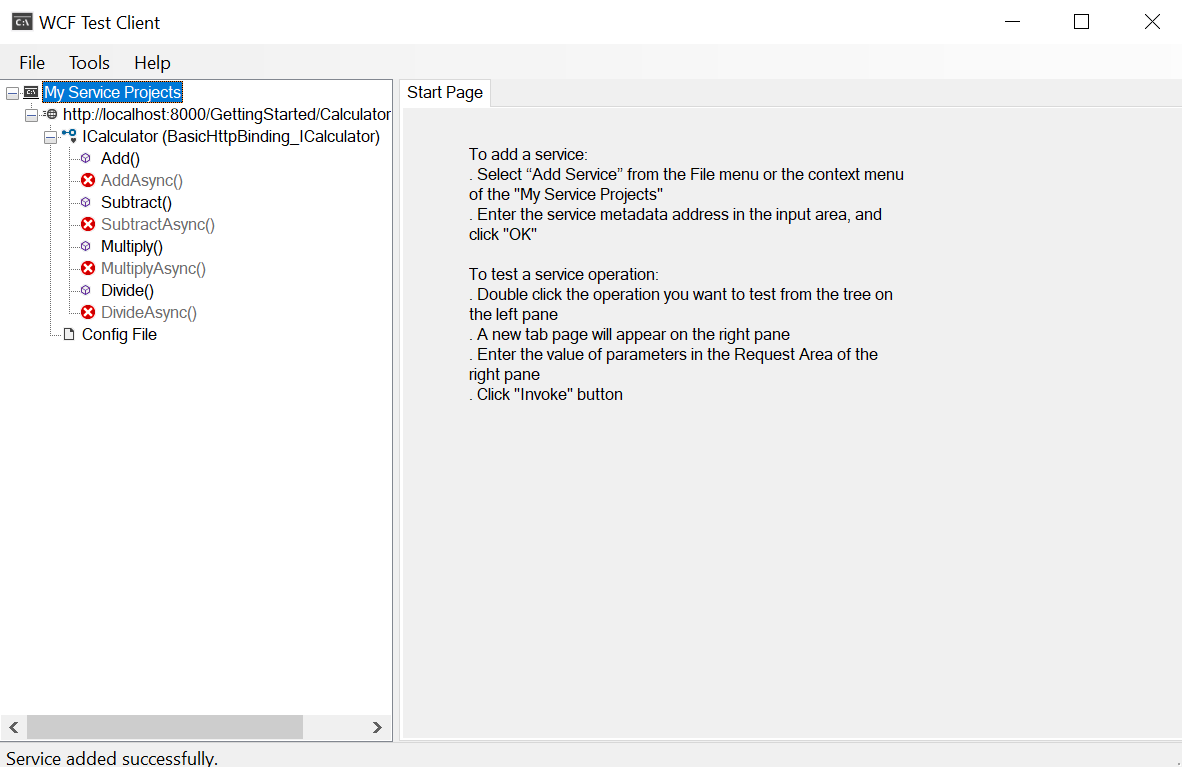
|  |
| --- |
| Dưới [<system.serviceModel>](https://docs.microsoft.com/en-us/dotnet/framework/configure-apps/file-schema/wcf/system-servicemodel), notice the [<endpoint>](https://docs.microsoft.com/en-us/dotnet/framework/configure-apps/file-schema/wcf/endpoint-element) element. The **<endpoint>** element defines the endpoint that the client uses to access the service as follows:   * Address: http://localhost:8000/GettingStarted/CalculatorService. The address of the endpoint. * Service contract: ServiceReference1.ICalculator. The service contract handles communication between the WCF client and the service. Visual Studio generated this contract when you used its **Add Service Reference** function. It's essentially a copy of the contract that you defined in the GettingStartedLib project. * Binding: [WSHttpBinding](https://docs.microsoft.com/en-us/dotnet/api/system.servicemodel.wshttpbinding). The binding specifies HTTP as the transport, interoperable security, and other configuration details. |

# Bước 5 - Sử dụng Client service

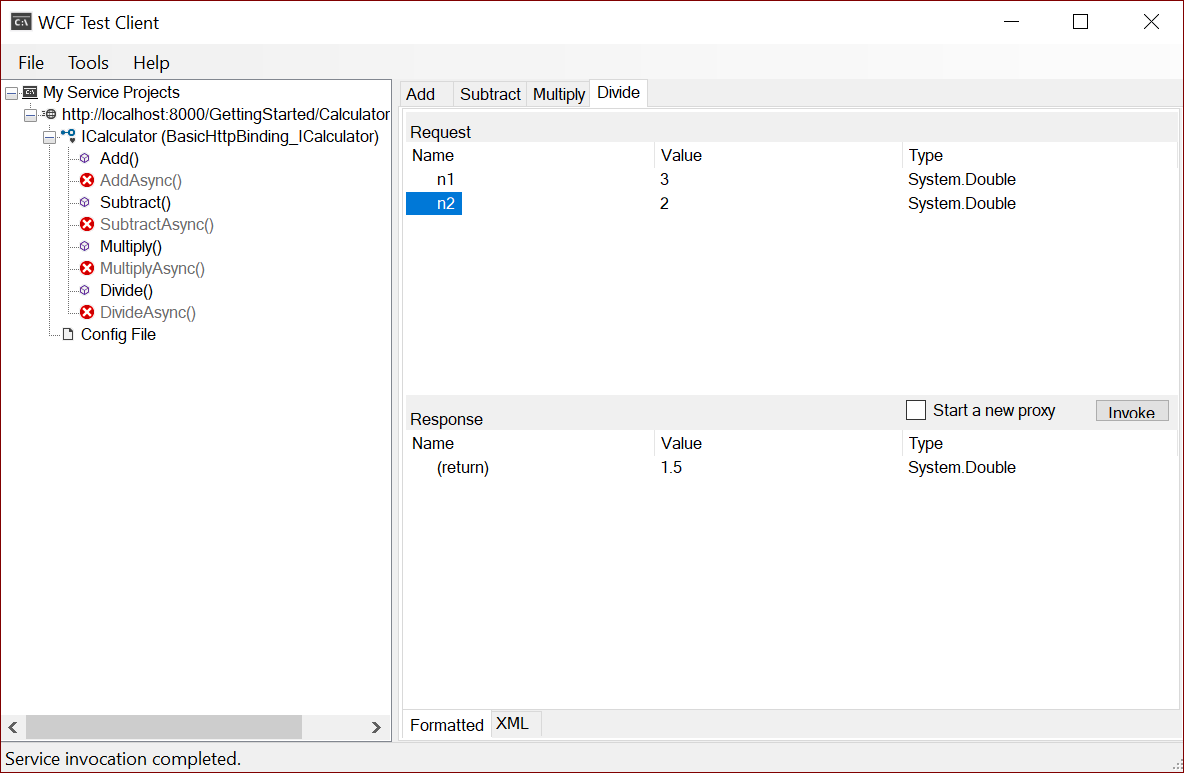
|  |
| --- |
| Add code to use the WCF client |
| class Program  {  static void Main(string[] args)  {  //Step 1: Create an instance of the WCF proxy.  CalculatorClient client = new CalculatorClient();  // Step 2: Call the service operations.  // Call the Add service operation.  double value1 = 100.00D;  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);  // Step 3: Close the client to gracefully close the connection and clean up resources.  Console.WriteLine("\nPress <Enter> to terminate the client.");  Console.ReadLine();  client.Close();  } } |

**Test thử dịch vụ trực tiếp trong Visual Studio**

* 1. Chọn GettingStartedLib > Set as Startup Project
  2. Bấm F5 để chạy dự án



Double click vào từng phương thức và nhập các giá trị sau đó bấm nut **Invoke** để xem kết quả



**Chạy từ dòng lệnh CMD**

1. Mở cửa sổ **command prompt** với quyền **administrator**, và chuyển đến thư mục chứa dự án
2. Để chạy dịch vụ, chạy: ***GettingStartedHost.exe*.**
3. Để chạy client, chạy: ***GettingStartedClient.exe*.**

