

Bài thực hành 05

Thực hành CRUD với ADO.NET

MỤC TIÊU

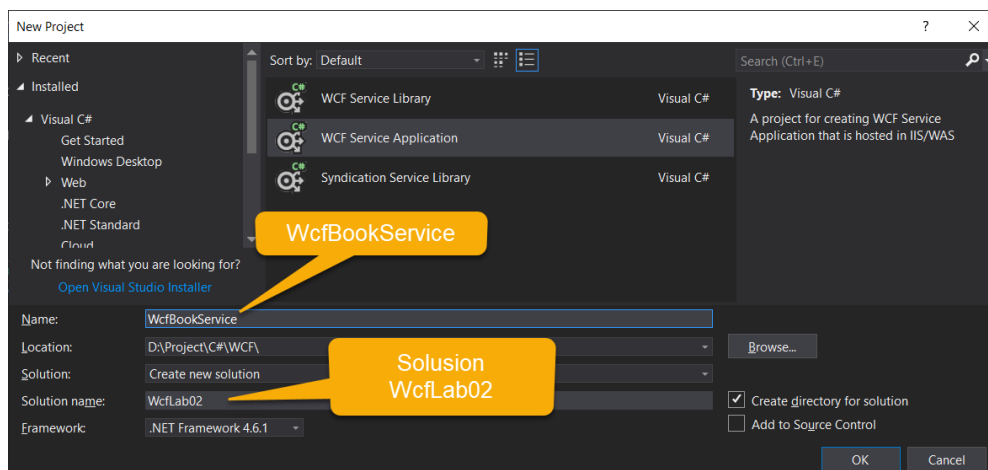
- Tạo WCF Services, kết nối CSDL Sql Server 2014..
- Sử dụng WCF Test Client để test các thao tác CRUD
- Cấu hình và host WCF Services
- Sử dụng WCF Services

BÀI 1: TẠO ỨNG DỤNG WCF SERVICE CUNG CẤP CÁC PHƯƠNG THỨC CRUD CƠ BẢN

Bước 1: Sinh viên tạo một CSDL mới đặt tên là **WcfBook**

```
create database WcfBook
go
use WcfBook
go
Create Table Author
(
    Id int Primary Key Identity,
    Name nvarchar(100) NOT NULL
)
GO
Create Table Book
(
    Id int Primary key Identity,
    Title nvarchar(120) NOT NULL,
    AuthorId int NOT NULL FOREIGN KEY REFERENCES Author(Id),
    Descriptions nvarchar(max) null,
    PosterLink varchar(200) NULL,
)
GO
INSERT INTO Author (Name) VALUES
(N'Ngô Tất Tố'),
(N'Vũ Trọng Phụng'),
(N'Tố Hữu'),
(N'Hồ Chí Minh'),
(N'Nam Cao')
```

Bước 2: Sinh viên tạo dự án WCF đặt tên là WcfBookService

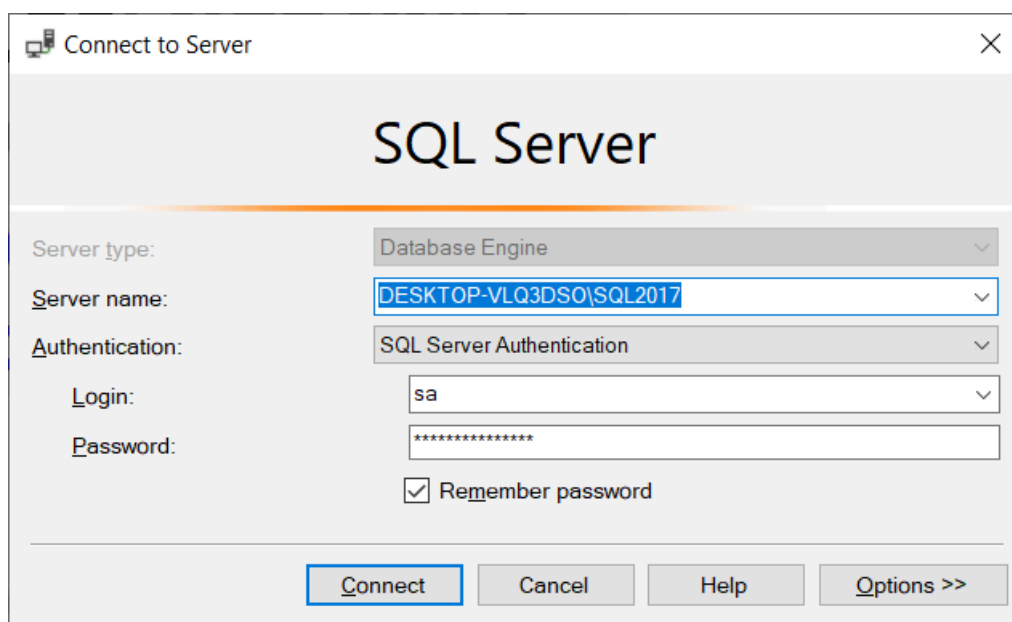


Bước 3: Cấu hình kết nối tới CSDL đã tạo bước 1

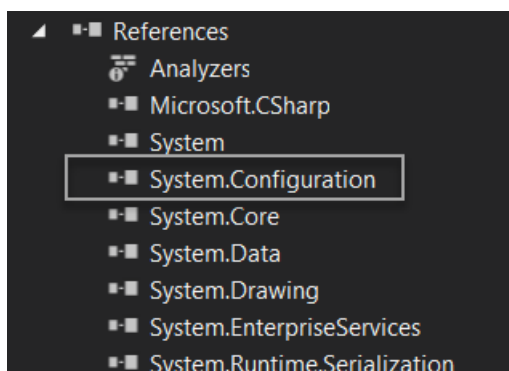
Mở file Web.config thêm mã cấu hình chuỗi kết nối có dạng

```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.6.1" />
    <httpRuntime targetFramework="4.6.1"/>
  </system.web>
  <!-- Thêm chuỗi kết nối cơ sở dữ liệu -->
  <connectionStrings>
    <add name="WcfBookConnStr" connectionString="Data Source=DESKTOP-VLQ3DS0\SQL2017;User Id=sa;Password=123a@;Database=WcfBook" />
  </connectionStrings>
  <!-- Các cấu hình khác để nguyên mặc định như ban đầu -->
</configuration>
```

Chú ý, Tên máy chủ của CSDL Data Source trong chuỗi kết nối trên có thể thấy khi khởi động Sql Server lên tại màn hình login

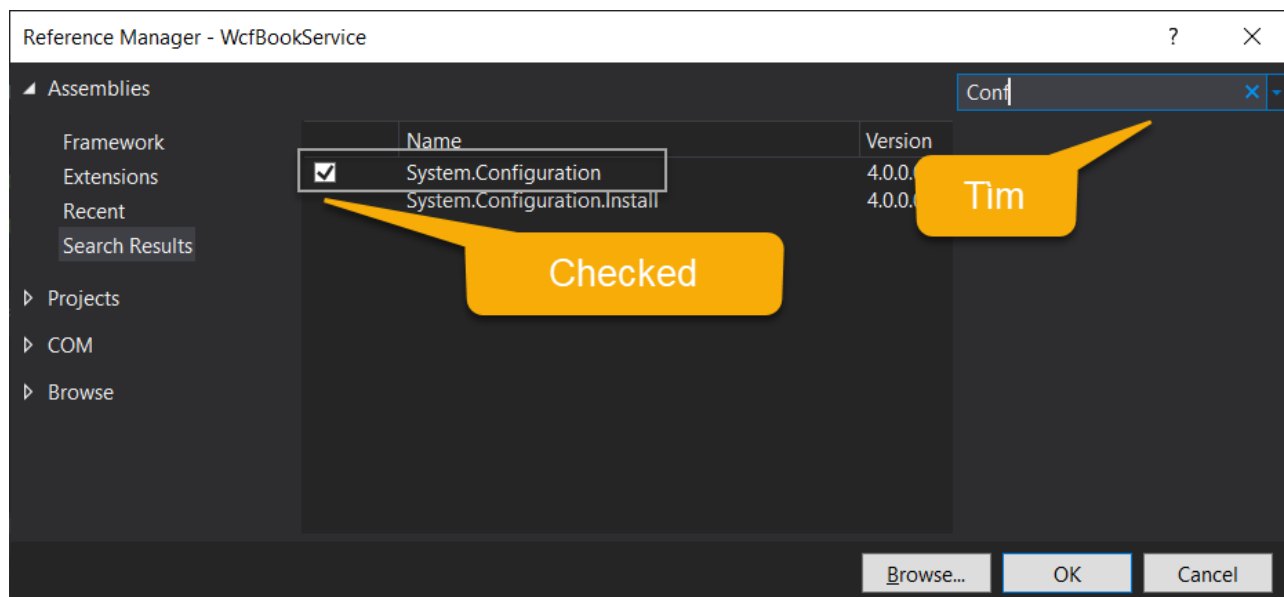


Bước 4: Kiểm tra trong thư mục References của dự án đã có thư System.Configuration chưa,



Nếu chưa thì add thêm vào bằng cách

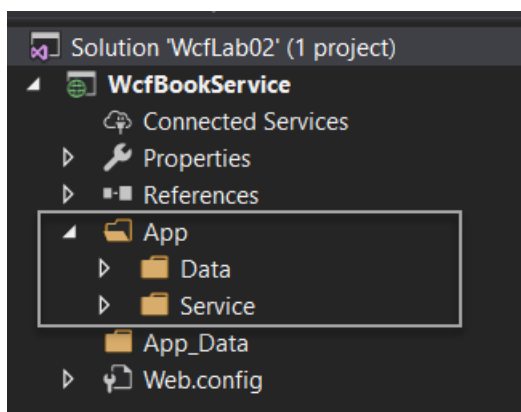
Click phải chuột lên thư mục References của dự án và chọn Add Reference...



Bước 5: Xóa các file *IService1.cs*, *Service1.svc* vì không cần thiết

Tạo thư mục App, trong App lại tạo thêm 3 thư mục, Data, Service

Cấu trúc dự án hiện tại có dạng



Bước 6: Trong thư mục Data tạo các class sau đây

Class **Db** (thực hiện mở và đóng kết nối)

```
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
namespace WcfBookService.App.Data
{
    public class Db
    {
        private static string conStr = ConfigurationManager.ConnectionStrings["WcfBoookConnStr"].ToString();
```

```
private static SqlConnection conn = new SqlConnection();
public static SqlCommand cmd = new SqlCommand("", conn);

public static void OpenConnection()
{
    if(conn.State == ConnectionState.Closed)
    {
        conn.ConnectionString = conStr;
        cmd.CommandType = CommandType.Text;
        conn.Open();
    }
}

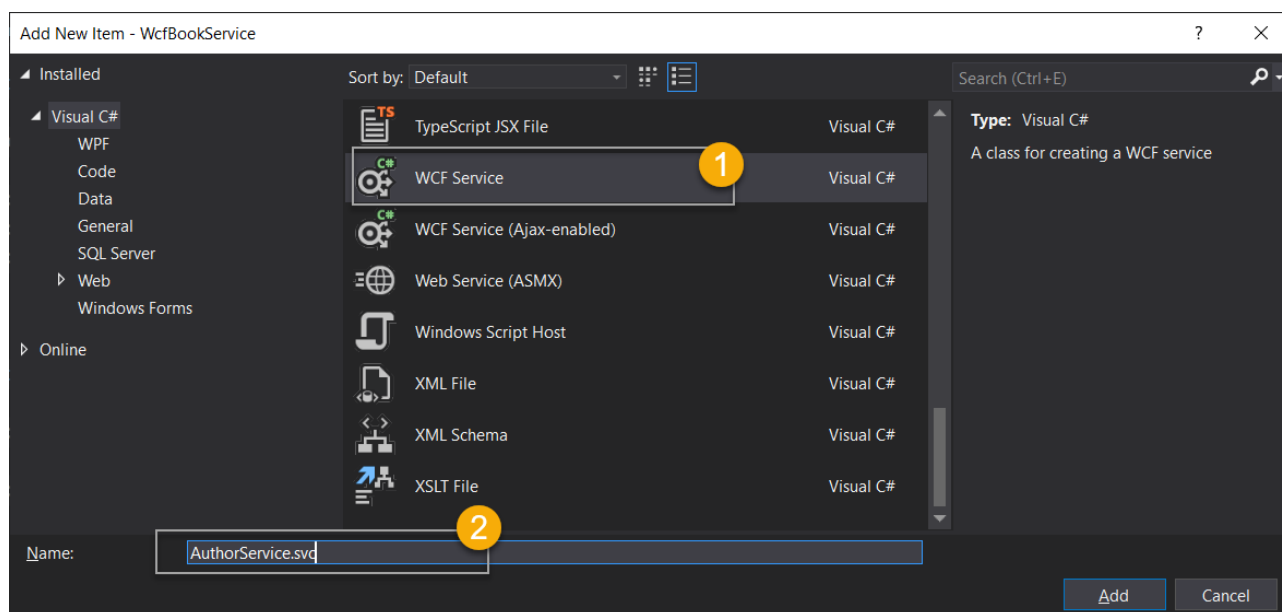
public static void CloseConnection()
{
    if (conn.State == ConnectionState.Open)
    {
        conn.Close();
    }
}
}
```

Class AuthorDataContract

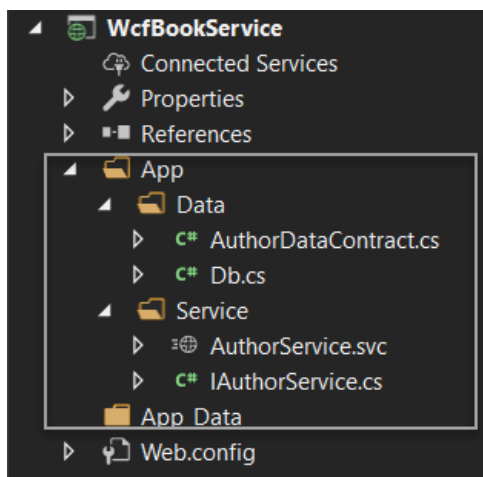
```
using System.Runtime.Serialization;
namespace WcfBookService.App.Data
{
    [DataContract]
    public class AuthorDataContract
    {
        [DataMember]
        public int Id { get; set; }
        [DataMember]
        public string Name { get; set; }
    }
}
```

Trong thư mục Service tạo AuthorService bằng cách

Click phải chuột lên thư mục Service chọn Add -> New item tìm đến WCF Service, đặt tên là **AuthorService**



Sau khi thêm, sẽ có 2 file là **IAuthorService.cs** và **AuthorService.svc**



Bước 7: Mở file IAuthorService lên và khai báo các phương thức cài đặt như sau

```
using System.Collections.Generic;
using System.ServiceModel;
using WcfBookService.App.Data;
namespace WcfBookService.App.Service
{
    [ServiceContract]
    public interface IAuthorService
    {
        [OperationContract]
        List<AuthorDataContract> GetAuthors();

        [OperationContract]
        AuthorDataContract GetAuthorById(int id);

        [OperationContract]
        int CreateNewAuthor(AuthorDataContract obj);

        [OperationContract]
        int UpdateAuthor(AuthorDataContract obj);

        [OperationContract]
        int DeleteAuthor(int id);
    }
}
```

Mở tiếp file AuthorService.svc và triển khai các phương thức đã cài đặt như sau

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using WcfBookService.App.Data;

namespace WcfBookService.App.Service
{
    public class AuthorService : IAuthorService
    {
        public int CreateNewAuthor(AuthorDataContract obj)
        {
            Db.OpenConnection(); // phương thức mở kết nối
            string sql = "INSERT INTO Author VALUES(@name)";
            Db.cmd.Parameters.AddWithValue("@name", obj.Name);
            Db.cmd.CommandText = sql;
            int check = Db.cmd.ExecuteNonQuery();
            Db.cmd.Parameters.Clear(); // Hủy bỏ các Parameters
            Db.CloseConnection(); // phương thức đóng kết nối
            return check;
        }
    }
}
```

```
}

public int DeleteAuthor(int id)
{
    Db.OpenConnection(); // phương thức mở kết nối
    string sql = "DELETE FROM Author WHERE Id = @id";
    Db.cmd.Parameters.AddWithValue("@id", id);
    Db.cmd.CommandText = sql;
    int check = Db.cmd.ExecuteNonQuery();
    Db.cmd.Parameters.Clear(); // Hủy bỏ các Parameters
    Db.CloseConnection(); // phương thức đóng kết nối
    return check;
}

public AuthorDataContract GetAuthorById(int id)
{
    AuthorDataContract author = new AuthorDataContract();
    Db.OpenConnection(); // phương thức mở kết nối
    string sql = "SELECT * FROM Author WHERE Id = @id";
    Db.cmd.Parameters.AddWithValue("@id", id);
    Db.cmd.CommandText = sql;
    SqlDataReader rd = Db.cmd.ExecuteReader();
    if (rd.HasRows && rd.Read())
    {
        author.Id = Convert.ToInt32(rd["Id"]);
        author.Name = rd["Name"].ToString();
    }
    Db.cmd.Parameters.Clear(); // Hủy bỏ các Parameters
    Db.CloseConnection(); // phương thức đóng kết nối
    return author;
}

public List<AuthorDataContract> GetAuthors()
{
    List<AuthorDataContract> authorsrs = new List<AuthorDataContract>();
    Db.OpenConnection(); // phương thức mở kết nối
    string sql = "SELECT * FROM Author Order By Name ASC";
    Db.cmd.CommandText = sql;
    SqlDataReader rd = Db.cmd.ExecuteReader();
    if (rd.HasRows)
    {
        while (rd.Read())
        {
            int id = Convert.ToInt32(rd["Id"]);
            string name = rd["Name"].ToString();
            authorsrs.Add(new AuthorDataContract { Id = id, Name = name });
        }
    }
    Db.CloseConnection(); // phương thức đóng kết nối
    return authorsrs;
}

public int UpdateAuthor(AuthorDataContract obj)
{
    Db.OpenConnection(); // phương thức mở kết nối
    string sql = "UPDATE Author SET Name = @name WHERE Id = @id";
    Db.cmd.Parameters.AddWithValue("@name", obj.Name);
    Db.cmd.Parameters.AddWithValue("@id", obj.Id);
    Db.cmd.CommandText = sql;
    int check = Db.cmd.ExecuteNonQuery();
    Db.cmd.Parameters.Clear(); // Hủy bỏ các Parameters
    Db.CloseConnection(); // phương thức đóng kết nối
    return check;
}
}
```

Chú ý: trong các phương thức trên chỗ nào có Parameter thì sau khi thực thi xong lệnh cần sử dụng Clear() để mỗi lần chạy lại thì các Parameter sẽ được nạp lại

Bước 8: Test dịch vụ đã tạo trên đây bằng cách



Click phải chuột lên file AuthorService.svc cho j Open in Browser

Dự án khởi động trên trình duyệt với link có dạng

<http://localhost:54872/App/Service/AuthorService.svc>

AuthorService 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:

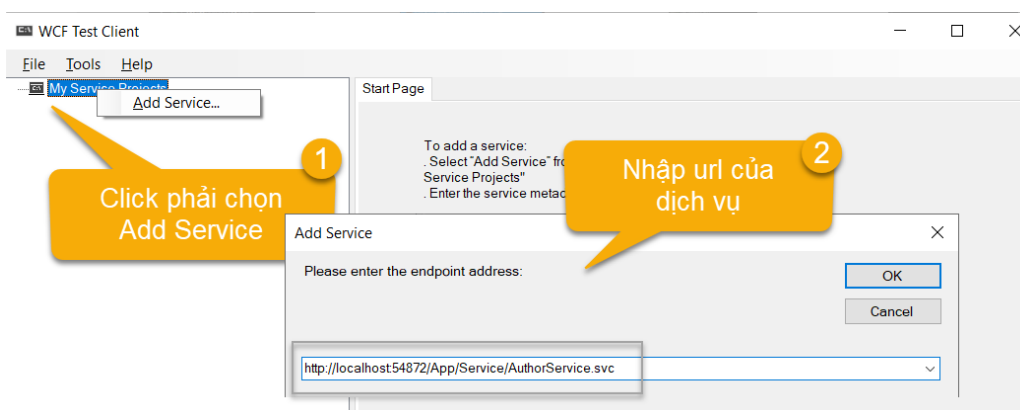
```
svcutil.exe http://localhost:54872/App/Service/AuthorService.svc?wsdl
```

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

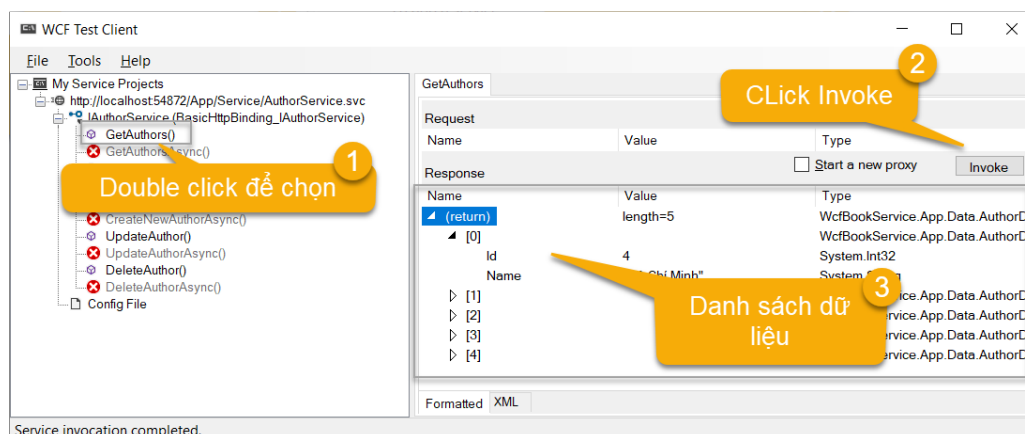
```
http://localhost:54872/App/Service/AuthorService.svc?singleWsdl
```

Mở ứng dụng **WCF Test Client** lên và Add Link trên vào

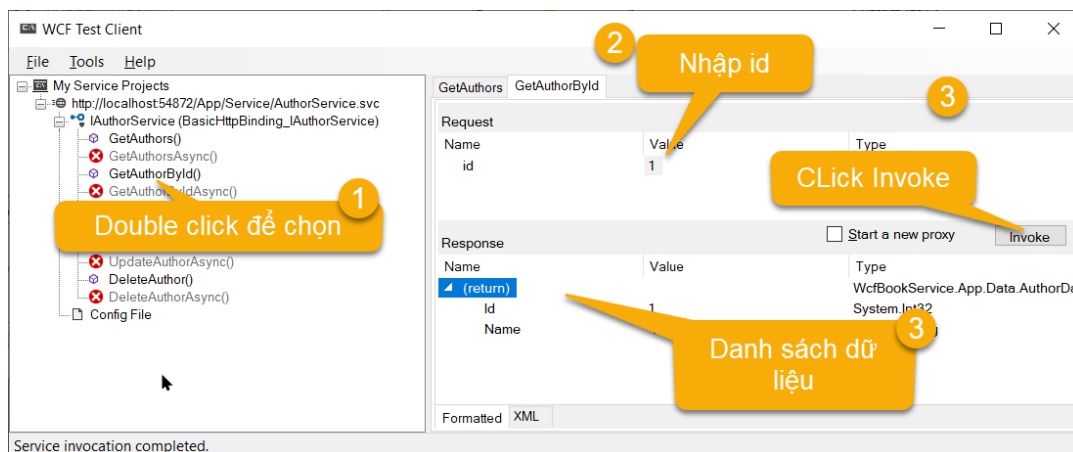
Chú ý ứng dụng WCF test Client mặc định đối với win 64bit trong thư mục **C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise\Common7\IDE\WcfTestClient.exe**



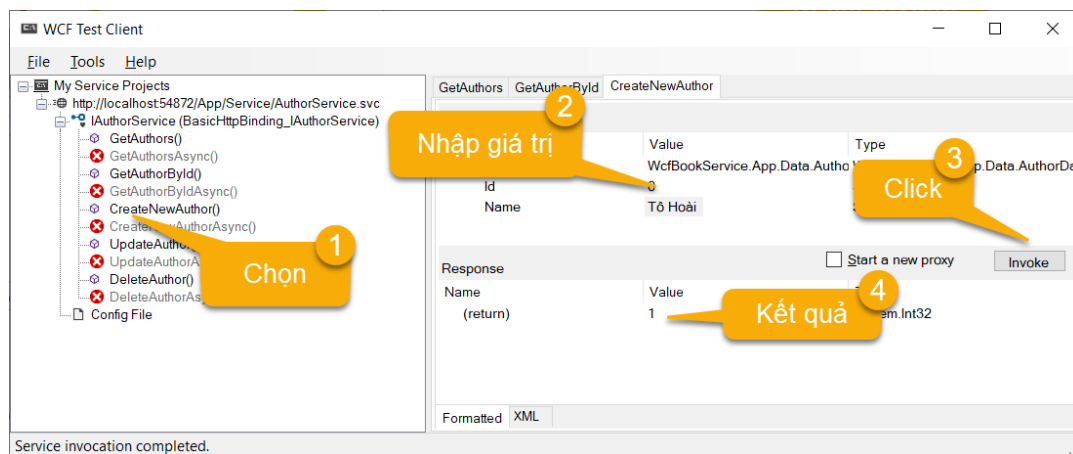
Test phương thức GetAuthors() trả về danh sách



Test phương thức GetAutorByld(int id) trả về một record theo tham số id



Test phương thức thêm mới CreateAuthor()



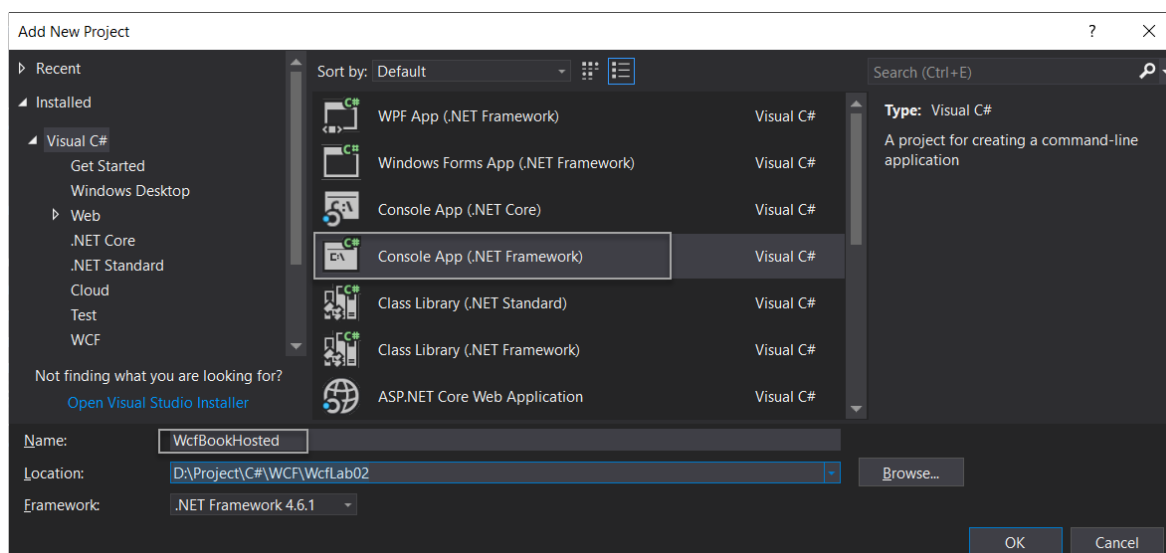
Tương tự với UpdateAuthor() cần nhập id của author cần sửa và giá trị mới cho Name

Phương thức DeleteAuthor(int id) cần truyền vào giá trị id của author cần xóa

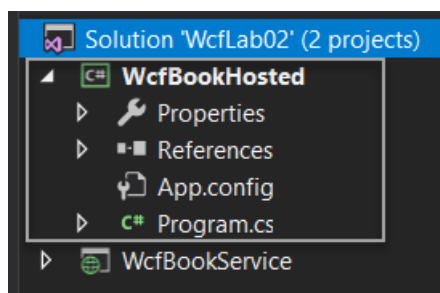
BÀI 2: CẤU HÌNH HOST VÀ ENDPOINT CHO ỨNG DỤNG WCF TRÊN ĐÂY

Bước 1: Trong Solution của dự án trên tạo thêm một Console App nữa đặt tên là WcfBookHosted, xem hình

Chú ý: sau khi tạo xong, click phải chuột lên dự án mới này chọn **Set as StartUp project** để cấu hình cho dự án này chạy chính của Solution

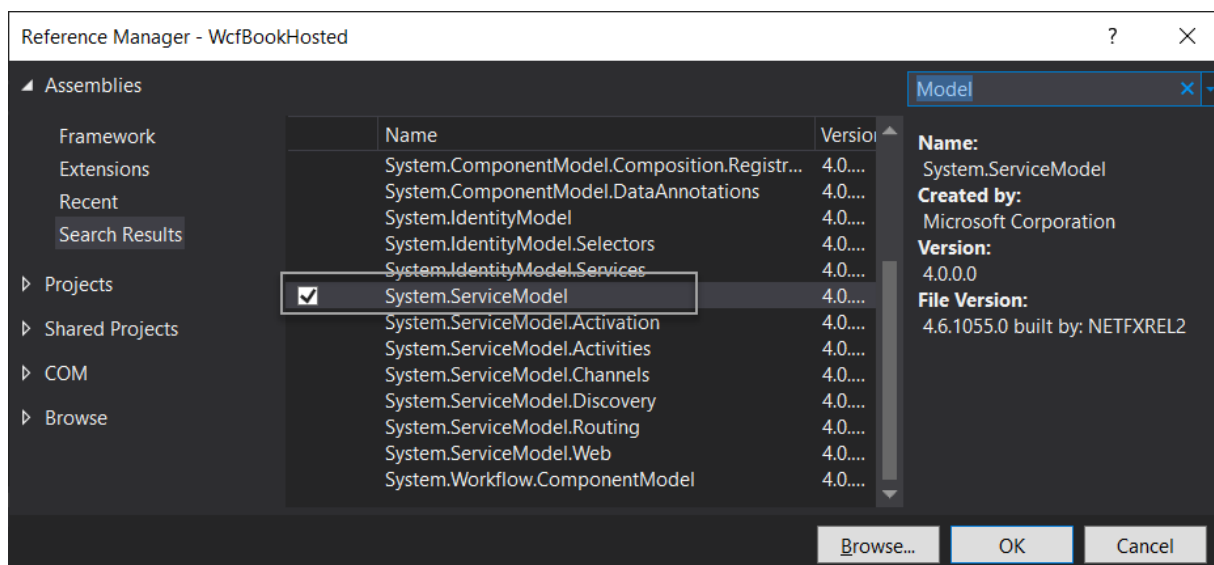


Cấu trúc dự án sau bước trên

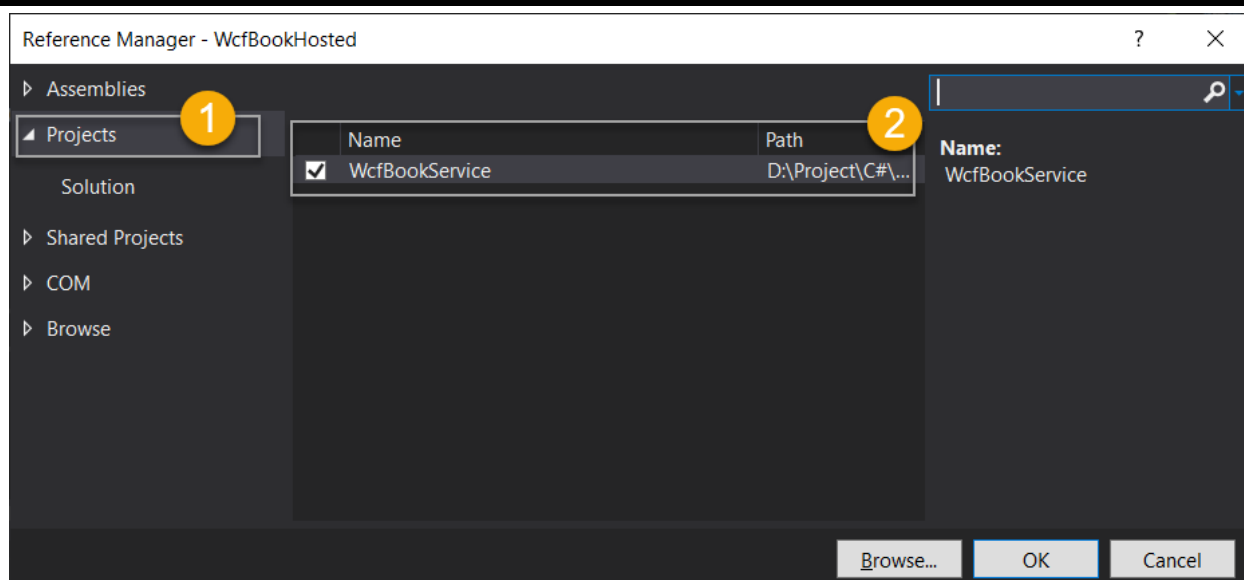


Bước 2: Add các References

System.ServiceModel: bằng cách click phải chuột lên thư mục References chọn Add Reference... tìm đến System.ServiceModel



WcfBookService bằng cách Click phải chuột lên References chọn Add Service Reference..., tìm Projects và chọn **WcfBookService**



Bước 3: Mở file App.config và copy chuỗi connectionStrings bên dự án WcfBookService sang dự án này

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
  <connectionStrings>
    <add name="WcfBoookConnStr" connectionString="Data Source=DESKTOP-
VLQ3DSO\SQL2017;Database=WcfBook;User Id=sa; Password=123a@" />
  </connectionStrings>
</configuration>
```

Bước 4: Tiếp tục tại file App.config cấu hình các services endpoint

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
  <connectionStrings>
    <add name="WcfBoookConnStr" connectionString="Data Source=DESKTOP-
VLQ3DSO\SQL2017;Database=WcfBook;User Id=sa; Password=123a@" />
  </connectionStrings>
  <system.serviceModel>
    <services>
      <service name="WcfBookService.App.Service.AuthorService">
        <host>
          <baseAddresses>
            <add baseAddress="http://localhost:99/AuthorService" />
          </baseAddresses>
        </host>
        <endpoint address="AuthorService" contract="WcfBookService.App.Service.IAuthorService"
binding="basicHttpBinding" />
      </service>
      <!-- Có thể tạo thêm các endpoint tại đây -->
    </services>
  </system.serviceModel>
</configuration>
```

Chú ý: Mỗi một endpoint được đặt trong cặp thẻ <service></service>

Như code trên đây có thể thấy



```
<service name="WcfBookService.App.Service.AuthorService">
  <host>
    <baseAddresses>
      <add baseAddress="http://localhost:99/AuthorService"/>
    </baseAddresses>
  </host>
  <endpoint address="AuthorService" contract="WcfBookService.App.Service.IAuthorService" binding="basicHttpBinding" />
</service>
```

Bước 5: Mở file Program.cs lên và soạn code mở host như sau

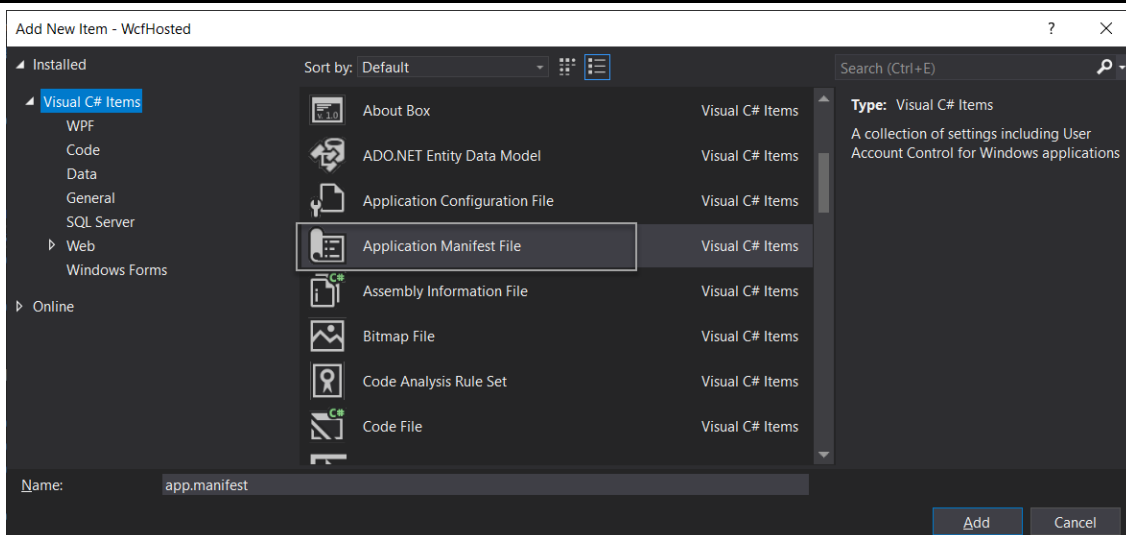
```
using System;
using System.ServiceModel;
using System.Text;
namespace WcfBookHosted
{
    class Program
    {
        static void Main(string[] args)
        {
            ServiceHost host = new ServiceHost(typeof(WcfBookService.App.Service.AuthorService));
            host.Open();
            Console.OutputEncoding = Encoding.UTF8; // hiển thị tiếng việt trên màn hình
            Console.WriteLine("-> MỞ CỘNG DỊCH VỤ ----->");
            Console.WriteLine("-> Hãy truy cập http://localhost:99/AuthorService");
            Console.ReadLine();
        }
    }
}
```

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

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 **WcfBookHosted** và chọn Add -> new Item và chọn đến **Application Manifest File**, để nguyên tên file là app.manifest

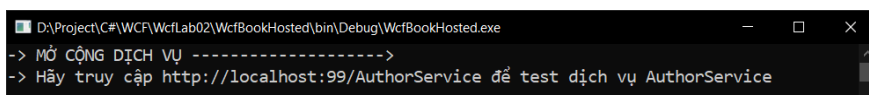


Sau đó tìm và cấu hình thuộc tính **requestedExecutionLevel** như sau

Nội dung toàn bộ file app.manifest có dạng

```
<?xml version="1.0" encoding="utf-8"?>
<assembly manifestVersion="1.0" xmlns="urn:schemas-microsoft-com:asm.v1">
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
    <security>
      <requestedPrivileges xmlns="urn:schemas-microsoft-com:asm.v3">
        <requestedExecutionLevel level="requireAdministrator" uiAccess="false" />
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
```

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



Truy cập link như màn hình thông báo sẽ có kết quả

Service

This is a Windows® Communication Foundation service.

Metadata publishing for this service is currently disabled.

If you have access to the service, you can enable metadata publishing by completing the following steps to modify your web or application configuration file:

Chú ý: MetaData chưa được publish

1. Create the following service behavior configuration, or add the <serviceMetadata> element to an existing service behavior configuration:

```
<behaviors>
  <serviceBehaviors>
    <behavior name="MyServiceTypeBehaviors" >
      <serviceMetadata httpGetEnabled="true" />
    </behavior>
  </serviceBehaviors>
</behaviors>
```

Bước 7: Cấu hình publish Metadata

Mở lại file App.config của dự án **WcfBookHosted**

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
  </startup>
  <connectionStrings>
    <add name="WcfBookConnStr" connectionString="Data Source=DESKTOP-VLQ3DS0\SQL2017;Database=WcfBook;User Id=sa; Password=123a@" />
  </connectionStrings>
  <system.serviceModel>
    <behaviors>
      <serviceBehaviors>
        <behavior name="WcfBookMetadata">
          <serviceMetadata httpGetEnabled="true"/>
        </behavior>
      </serviceBehaviors>
    </behaviors>
    <services>
      <service name="WcfBookService.App.Service.AuthorService" behaviorConfiguration="WcfBookMetadata">
        <host>
          <baseAddresses>
            <add baseAddress="http://localhost:99/AuthorService"/>
          </baseAddresses>
        </host>
        <endpoint address="AuthorService" contract="WcfBookService.App.Service.IAuthorService" binding="basicHttpBinding" />
      </service>
    </services>
  </system.serviceModel>
</configuration>
```

Khởi động lại dự án và truy cập link như trên kết quả sẽ có chút khác

AuthorService Service

You have created a service.

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

Metadata đã publish

```
svcutil.exe http://localhost:99/AuthorService?wsdl
```

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

```
http://localhost:99/AuthorService?singleWsd1
```

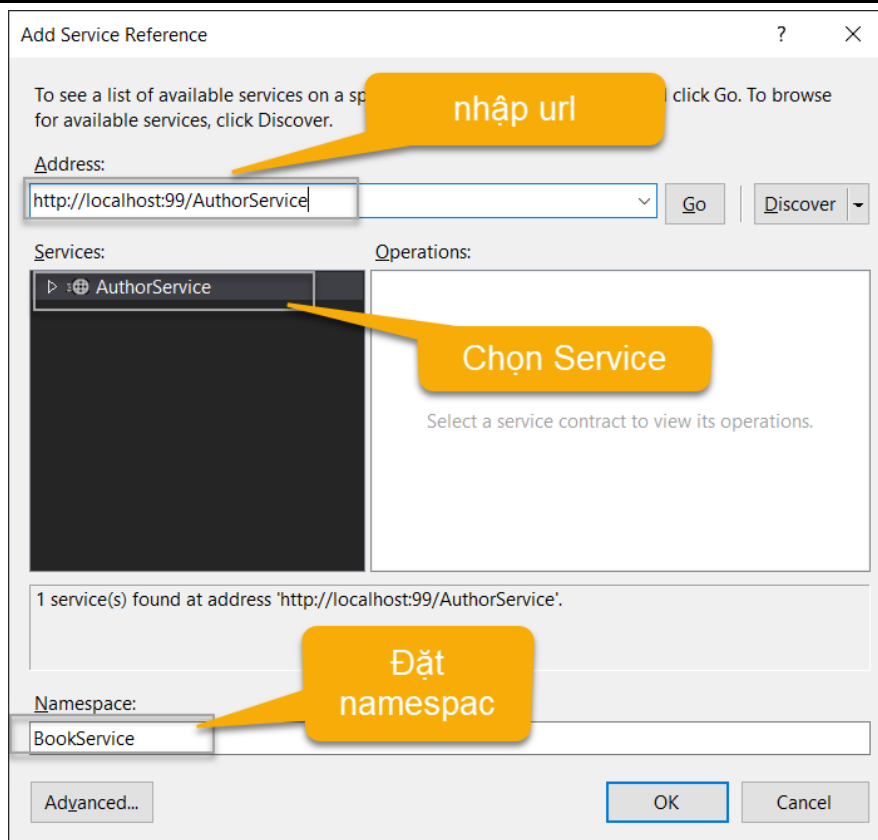
This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

BÀI 3: TẠO DỰ ÁN ASP.NET MVC ĐỂ TEST DỊCH VỤ TRÊN

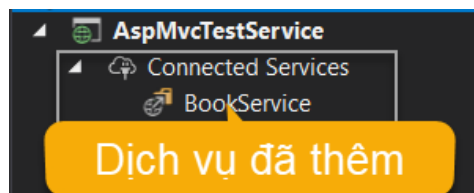
Bước 1: Mở thêm Visual Studio 2017 lên tạo dự án .NET MVC mới đặt tên là AspMvcTestService

Bước 2: Tiếp theo sau khi tạo dự án, click chuột phải lên thư mục References và chọn Add Service Reference...

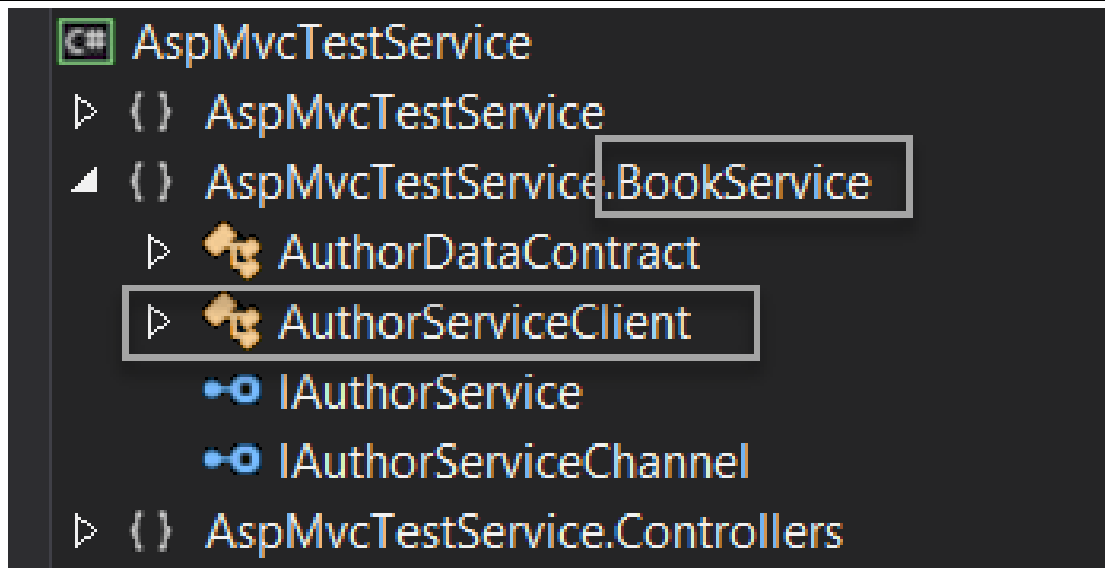
Sau khi màn hình sau đây sẽ hiển thị hãy nhập url của dịch vụ đã khởi động ở trên
Theo hình sau



Kết quả sau khi thêm dịch vụ có namespace tương ứng sẽ được thêm vào



Click vào namespace trên sẽ thấy các thành phần được sinh ra tương ứng như sau



Bước 3: Tạo một Controller mới đặt tên là AuthorController

Tại ActionResult Index code C# sử dụng BookService như sau

```
using AspMvcTestService.BookService;
using System;
using System.Web.Mvc;

namespace AspMvcTestService.Controllers
{
    public class AuthorController : Controller
    {
        private AuthorServiceClient client = new AuthorServiceClient();
        public ActionResult Index()
        {
            var authors = client.GetAuthors();
            ViewBag.authors = authors;
            client.Close();
            return View();
        }
    }
}
```

Bước 4: Add view cho action Index ở trên và code như sau

```
@{
    ViewBag.Title = "Author Page";
}

<h2>Danh sách các tác giả</h2>
<a href="@Url.Action("Create","Author")" class="btn btn-primary">Thêm mới</a>
<table class="table table-hover">
    <thead>
        <tr>
            <th>Author Id</th>
            <th>Author Name</th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        @foreach (var author in ViewBag.authors)
        {
            <tr>
```



```
<td>@author.Id</td>
<td>@author.Name</td>
<td>
    <a href="@Url.Action("Detail","Author", new { id = author.Id})" class="btn btn-sm btn-
primary">Xem chi tiết</a>
    <a href="@Url.Action("Edit","Author", new { id = author.Id})" class="btn btn-sm btn-
success">Sửa</a>
    <a href="@Url.Action("Delete","Author", new { id = author.Id})" class="btn btn-sm btn-
danger" onclick="return confirm('Bạn có chắc muốn xóa')">Xóa</a>
</td>
</tr>
}

</tbody>
</table>
```

Build lại dự án này và truy cập author link xem kết quả

<http://localhost:56816/Author>

Application name Home About Contact			
Danh sách các tác giả			
Author Id	Author Name		
4	Hồ Chí Minh	Xem chi tiết	Sửa Xóa
5	Nam Cao	Xem chi tiết	Sửa Xóa
1	Ngô Tất Tố	Xem chi tiết	Sửa Xóa
7	Tô Hoài	Xem chi tiết	Sửa Xóa
3	Tố Hữu	Xem chi tiết	Sửa Xóa
2	Vũ Trọng Phụng	Xem chi tiết	Sửa Xóa

Bước 5: Để tạo chức năng thêm mới, mở lại AuthorController tạo 2 action

Action Create với phương thức GET để hiển thị form thêm mới

Action Create với phương thức POST để nhận dữ liệu từ form

```
using AspMvcTestService.BookService;
using System;
using System.Web.Mvc;

namespace AspMvcTestService.Controllers
{
    public class AuthorController : Controller
    {
        // khai báo dịch vụ BookService
        private AuthorServiceClient client = new AuthorServiceClient();
        public ActionResult Index()
        {
            var authors = client.GetAuthors();
            ViewBag.authors = authors;
            client.Close();
        }
    }
}
```

```
        return View();
    }

    [HttpGet]
    public ActionResult Create()
    {
        return View();
    }

    [HttpPost]
    public ActionResult Create(FormCollection collection)
    {
        string name = collection["Name"]; // Lấy dữ liệu trên form
        AuthorDataContract author = new AuthorDataContract { Id = 0, Name = name };
        int isCreated = client.CreateNewAuthor(author); // Gọi phương thức thêm mới của AuthorService
        client.Close();
        return RedirectToAction("Index", "Author");
    }
}
```

Bước 6: Add view cho Action Create và soạn form như sau

```
@{
    ViewBag.Title = "Author Create";
}

<h2>Thêm mới</h2>
<form action="" method="post">
    <div class="row">
        <div class="col-md-4">
            <div class="form-group">
                <label>Tên tác giả</label>
                <input name="Name" class="form-control" placeholder="Tên tác giả..." />
            </div>
            <button class="btn btn-sm btn-primary">Lưu lại</button>
        </div>
    </div>
</form>
```

Build lại dự án, truy cập link thêm mới có dạng

<http://localhost:56816/Author/Create>

Thêm mới

Tên tác giả

Lưu lại

Hãy nhập dữ liệu và test thử thêm mới

Bước 7: Trong AuthorController Tạo thêm Action Edit

Action Edit với phương thức GET để hiển thị form thêm mới

Action Edit với phương thức POST để nhận dữ liệu từ form



```
using AspMvcTestService.BookService;
using System;
using System.Web.Mvc;
namespace AspMvcTestService.Controllers
{
    public class AuthorController : Controller
    {
        private AuthorServiceClient client = new AuthorServiceClient();
        // Các action khác vẫn để ở đây

        [HttpGet]
        public ActionResult Edit(int id)
        {
            var author = client.GetAuthorById(id); // truy cập dữ liệu từ AuthorService
            ViewBag.author = author; // Gửi dữ liệu qua view
            client.Close();
            return View();
        }

        [HttpPost]
        public ActionResult Edit(FormCollection collection)
        {
            int id = Convert.ToInt32(collection["Id"]);
            string name = collection["Name"]; // Lấy dữ liệu trên form
            AuthorDataContract author = new AuthorDataContract { Id = id, Name = name};
            int isUpdated = client.UpdateAuthor(author); // gọi phương thức update
            client.Close();
            return RedirectToAction("Index", "Author");
        }
    }
}
```

Bước 8: Add View Edit có code như sau

```
@{
    ViewBag.Title = "Author Edit";
    var author = ViewBag.author;
}
<h2>Chỉnh sửa</h2>
<form action="" method="post">
    <input type="hidden" name="id" value="@author.Id" />
    <div class="row">
        <div class="col-md-4">
            <div class="form-group">
                <label>Tên tác giả</label>
                <input name="Name" class="form-control" value="@author.Name" />
            </div>
            <button class="btn btn-sm btn-primary">Lưu lại</button>
        </div>
    </div>
</form>
```

Build lại dự án và click vào nút Edit trên danh sách để đi đến form Edit link có dạng

Chỉnh sửa

Tên tác giả



Bước 9: Thêm Action Detail

Code C#

```
using AspMvcTestService.BookService;
using System;
using System.Web.Mvc;
namespace AspMvcTestService.Controllers
{
    public class AuthorController : Controller
    {
        private AuthorServiceClient client = new AuthorServiceClient();
        // Các action throws đó vẫn để ở đây

        public ActionResult Detail(int id)
        {
            var author = client.GetAuthorById(id);
            ViewBag.author = author;
            client.Close();
            return View();
        }
    }
}
```

Code trên view

```
@{
    ViewBag.Title = "Author Detail";
    var author = ViewBag.author;
}

<h2>Chi tiết tác giả</h2>

<table class="table table-hover">
    <tbody>
        <tr>
            <td>Author Id</td>
            <td>@author.Id</td>
        </tr>
        <tr>
            <td>Author Name</td>
            <td>@author.Name</td>
        </tr>
    </tbody>
</table>

<a href="@Url.Action("Index", "Author")" class="btn btn-sm btn-primary">Về danh sách</a>
<a href="@Url.Action("Edit", "Author", new { id = author.Id})" class="btn btn-sm btn-success">Sửa</a>
<a href="@Url.Action("Delete", "Author", new { id = author.Id})" class="btn btn-sm btn-danger" onclick="return confirm('Bạn có chắc muốn xóa')">Xóa</a>
```

Kết quả, khi click lên nút bấm Xem chi tiết link có dạng

<http://localhost:56816/Author/Detail/9>



Chi tiết tác giả

Author Id	9
Author Name	Cao Bá Quát

Về danh sách

Sửa

Xóa

Bước 10: Tạo Action Delete thực hiện xóa, Code C#

```
using AspMvcTestService.BookService;
using System;
using System.Web.Mvc;

namespace AspMvcTestService.Controllers
{
    public class AuthorController : Controller
    {
        private AuthorServiceClient client = new AuthorServiceClient();
        // Các action throwcs đó vẫn để ở đây
        public ActionResult Delete(int id)
        {
            int isDeleted = client.DeleteAuthor(id);
            client.Close();
            return RedirectToAction("Index", "Author");
        }
    }
}
```

Build lại dự án và click và click vào nút xóa trên danh sách, link có dạng

<http://localhost:56816/Author/Delete/9>

Toàn bộ code của AuthorController

```
using AspMvcTestService.BookService;
using System;
using System.Web.Mvc;

namespace AspMvcTestService.Controllers
{
    public class AuthorController : Controller
    {
        private AuthorServiceClient client = new AuthorServiceClient();

        public ActionResult Index()
        {
            var authors = client.GetAuthors();
            ViewBag.authors = authors;
            client.Close();
            return View();
        }

        [HttpGet]
        public ActionResult Create()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Create(FormCollection collection)
        {
            string name = collection["Name"];
            AuthorDataContract author = new AuthorDataContract { Id = 0, Name = name };
            int isCreated = client.CreateNewAuthor(author);
            client.Close();
            return RedirectToAction("Index", "Author");
        }
    }
}
```



```
}

public ActionResult Detail(int id)
{
    var author = client.GetAuthorById(id);
    ViewBag.author = author;
    client.Close();
    return View();
}

[HttpGet]
public ActionResult Edit(int id)
{
    var author = client.GetAuthorById(id);
    ViewBag.author = author;
    client.Close();
    return View();
}

[HttpPost]
public ActionResult Edit(FormCollection collection)
{
    int id = Convert.ToInt32(collection["Id"]);
    string name = collection["Name"];
    AuthorDataContract author = new AuthorDataContract { Id = id, Name = name};
    int isUpdated = client.UpdateAuthor(author);
    client.Close();
    return RedirectToAction("Index", "Author");
}

public ActionResult Delete(int id)
{
    int isDeleted = client.DeleteAuthor(id);
    client.Close();
    return RedirectToAction("Index", "Author");
}
}
```

BÀI TẬP TỰ LÀM

Sinh viên thực hiện bảng Book còn lại trong CSDL trên đây để hoàn thiện chức năng CRUD

```
create database WcfBook
go
use WcfBook
go
Create Table Author
(
    Id int Primary Key Identity,
    Name nvarchar(100) NOT NULL
)
GO
Create Table Book
(
    Id int Primary key Identity,
    Title nvarchar(120) NOT NULL,
    AuthorId int NOT NULL FOREIGN KEY REFERENCES Author(Id),
    Descriptions nvarchar(max) null,
    PosterLink varchar(200) NULL,
)
GO
INSERT INTO Author (Name) VALUES
(N'Ngô Tất Tố'),
(N'Vũ Trọng Phụng'),
(N'Tố Hữu'),
(N'Hồ Chí Minh'),
(N'Nam Cao')
```

