

Bài thực hành 05.1

Thực hành CRUD với Entity Framework

MỤC TIÊU

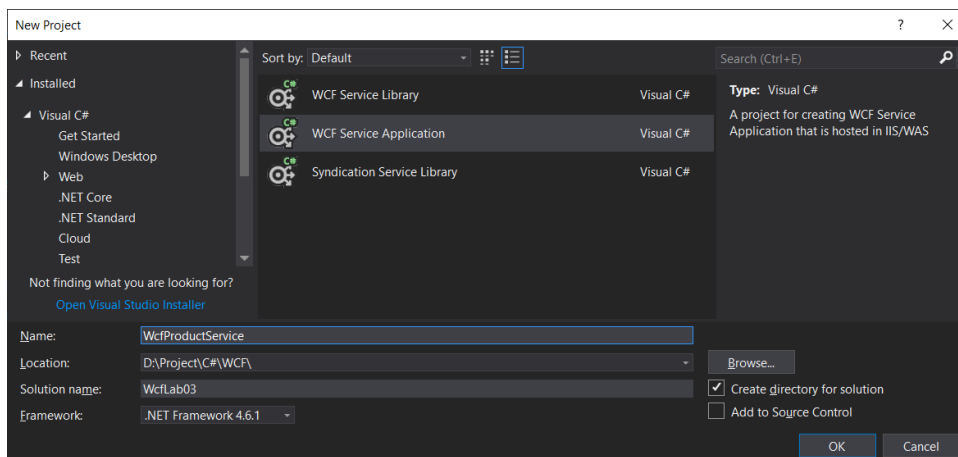
- Tạo WCF Services, kết nối CSDL Sql Server 2014..
- Sử dụng WCF Test Client để test các thao tác CRUD
- Cấu hình và host WCF Services
- Sử dụng WCF Services

BÀI 1: TẠO ỨNG DỤNG WCF SERVICE CRUD VỚI ENTITY

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

```
create database WcfProduct
go
use WcfProduct
go
Create Table Category
(
    Id int Primary Key Identity,
    Name nvarchar(100) NOT NULL
)
GO
Create Table Product
(
    Id int Primary key Identity,
    Title nvarchar(120) NOT NULL,
    Price float NOT NULL,
    SalePrice float NOT NULL,
    CategoryId int NOT NULL FOREIGN KEY REFERENCES Category(Id),
    Descriptions nvarchar(max) null,
    ImageLink varchar(200) NULL,
)
GO
INSERT INTO Category(Name) VALUES
(N'Quần Áo'),
(N'Túi Xách'),
(N'Đồng Hồ'),
(N'Máy Bơm'),
(N'Máy Sinh Tố')
```

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

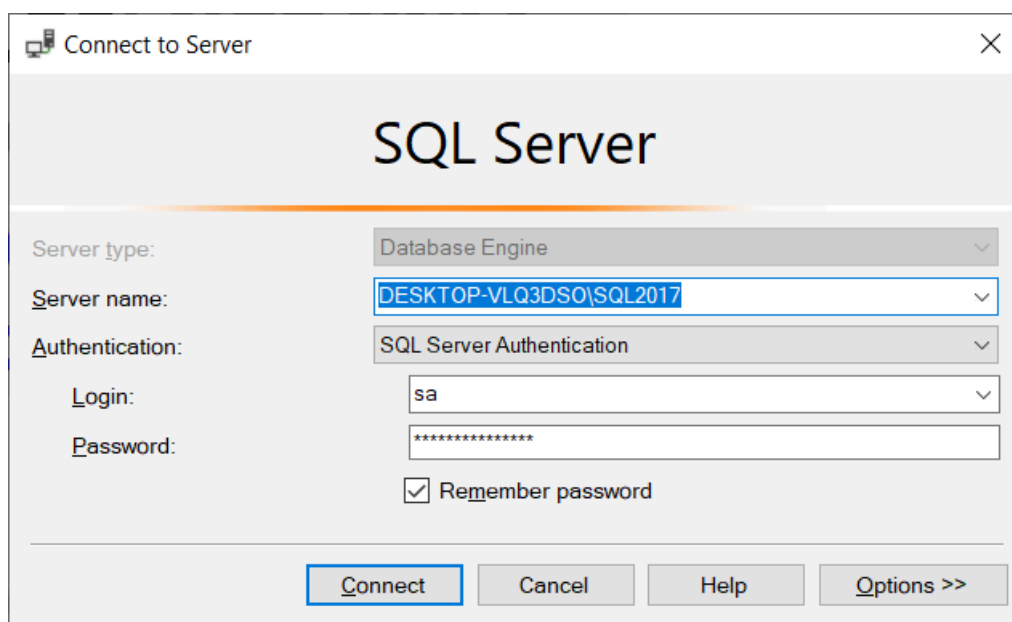
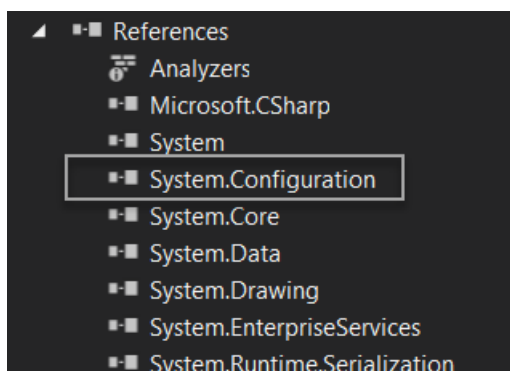


Bước 3: Cấu hình kết nối tới CSDL đã tạo bước 1

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

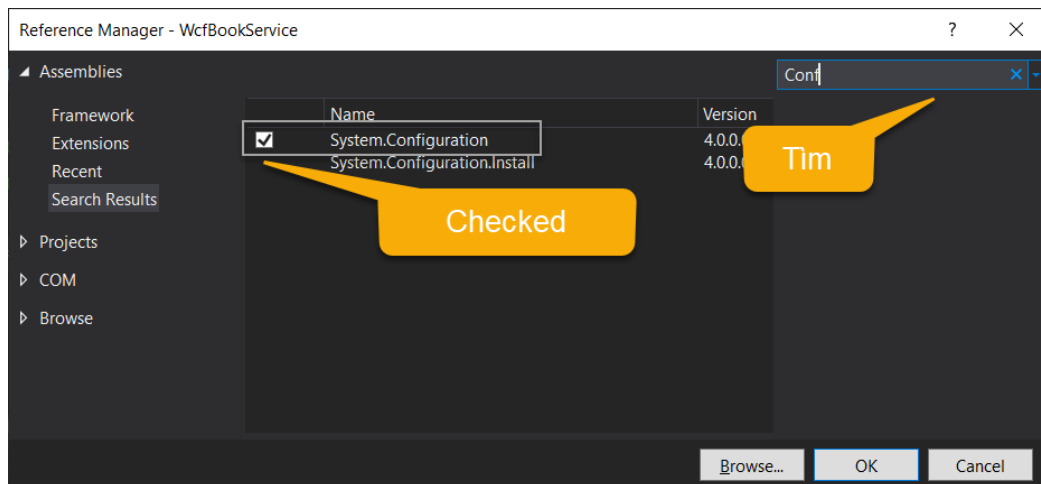
```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.6.1" />
    <httpRuntime targetFramework="4.6.1"/>
  </system.web>
  <!-- Thêm chuỗi kết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

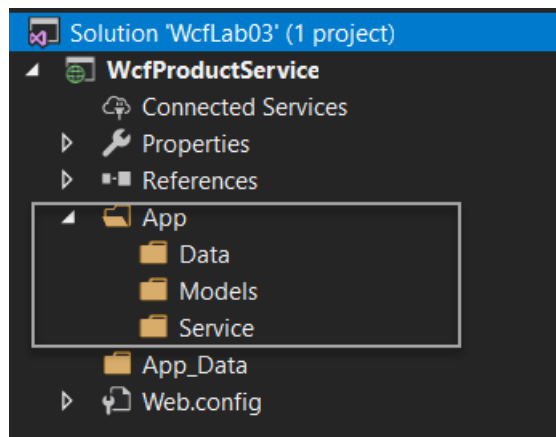
Click phải chuột lên thư mục References của dự án và chọn Add Reference...



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, Models, Service

Cấu trúc dự án hiện tại có dạng



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

Class **CategoryDataContract**

```
using System.Runtime.Serialization;

namespace WcfProductService.App.Data
{
    [DataContract]
    public class CategoryDataContract
    {
        [DataMember]
        public int Id { get; set; }
        [DataMember]
        public string Name { get; set; }
    }
}
```

Class **CategoryDataContract**

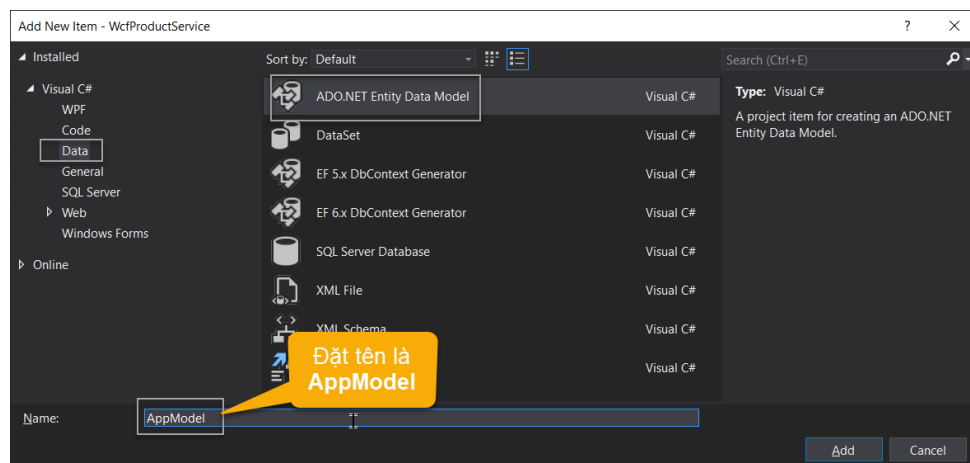
```
using System.Runtime.Serialization;

namespace WcfProductService.App.Data
{
    [DataContract]
    public class ProductDataContract
    {
        [DataMember]
        public int Id { get; set; }
        [DataMember]
        public string Title { get; set; }
        [DataMember]
        public double Price { get; set; }
        [DataMember]
        public double SalePrice { get; set; }
        [DataMember]
        public int CategoryId { get; set; }
        [DataMember]
        public string Descriptions { get; set; }
        [DataMember]
        public string ImageLink { get; set; }
    }
}
```

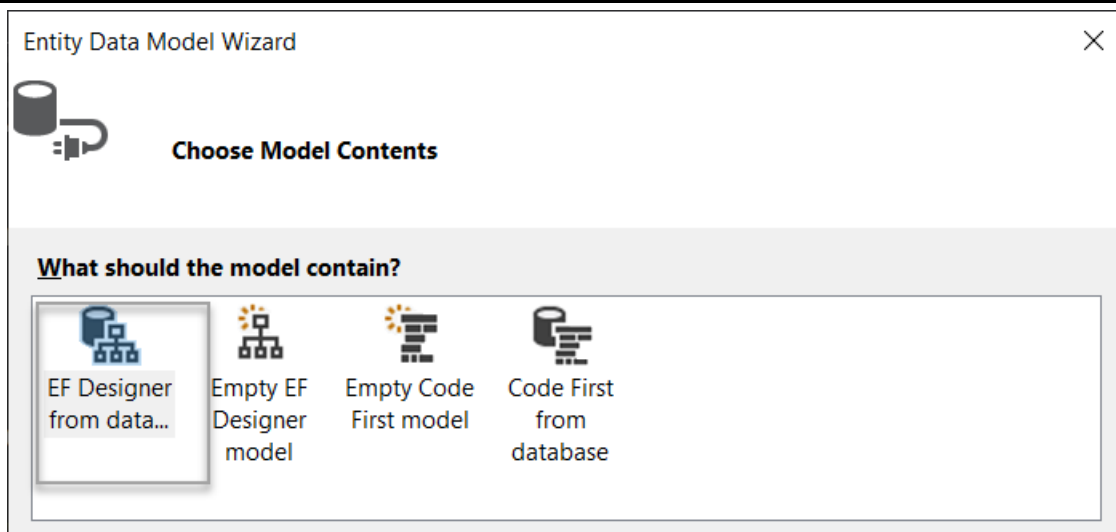
Bước 7: Tạo các Wcf Service sau

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

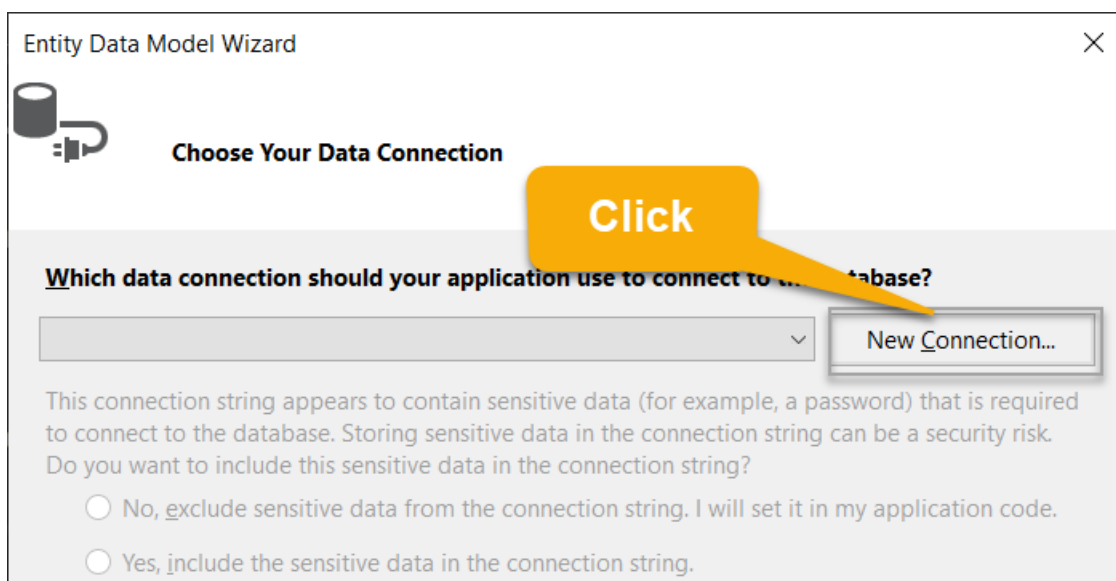
Bước 8: Click phải chuột lên thư mục Models Add -> New Item, tìm đến Entity Data Model và thực hiện lần lượt theo các hình ảnh sau



Màn hình tiếp theo chọn EF Designer from data...



Màn hình tiếp theo chọn vào Create Connection



Màn hình tiếp theo cấu hình kết nối CSDL WcfProduct

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
DESKTOP-VLQ3DSO\SQL2017 Refresh

Log on to the server

Authentication: SQL Server Authentication

User name: sa

Password: Save my password

Connect to a database

☒ Select or enter a database name:
WcfProduct

☐ Attach a database file: Browse...

Logical name: Advanced...

Test Connection OK Cancel

Màn hình tiếp theo

Entity Data Model Wizard

Choose Your Data Connection

Which data connection should I use to connect to the database?

desktop-vlq3dso\sql2017.W

New Connection...

This connection string appears to be a valid connection string. It may contain sensitive information (such as a password) that is required to connect to the database. Including this information in the connection string can be a security risk. Do you want to include the sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☒ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/App.Models.AppModel.csdl|res://*/App.Models.AppModel.ssdl|res://*/App.Models.AppModel.msl;provider=System.Data.SqlClient;provider connection string="data source=DESKTOP-VLQ3DSO\SQL2017;initial catalog=WcfProduct;user id=sa;password=*****;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Save connection settings in Web.Config

MyConnection

< Previous Next > Finish Cancel

Thêm chuỗi kết nối vào config

Tên chuỗi kết nối

Qua vài màn hình đến màn hình sau

Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

☒ Tables

☒ dbo

☒ Category

☒ Product

☐ Views

☐ Stored Procedures and Functions

☐ Pluralize or singularize generated object names

☒ Include foreign key columns in the model

☐ Import selected stored procedures into the model

Model Namespace:

AppModel

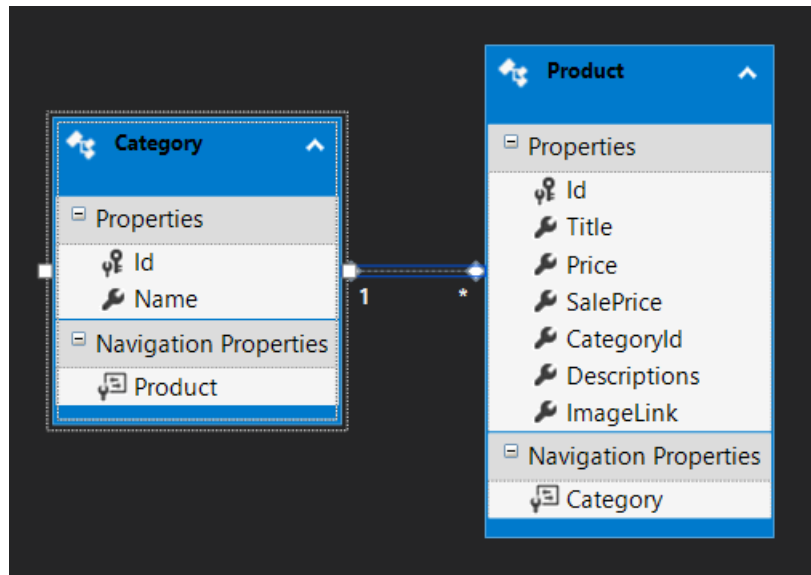
< Previous Next > Finish Cancel

Chọn các bảng

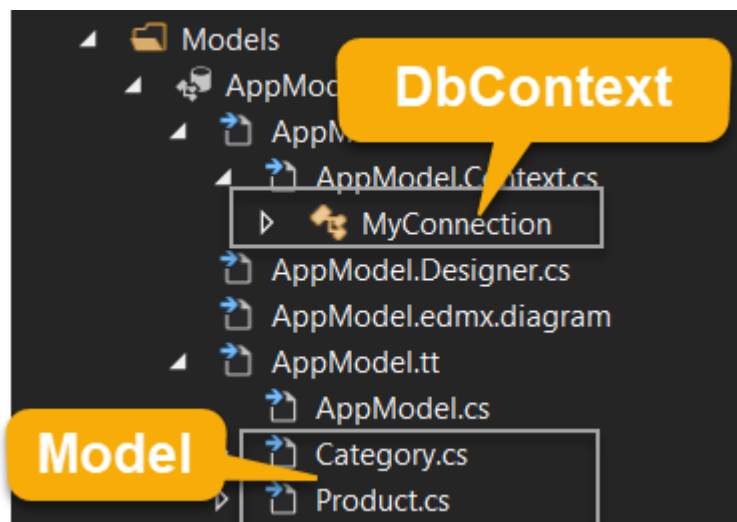
Tên model

Sau khi kết thúc, phần mềm sẽ tạo ra các thành phần sau

Entity Diagram



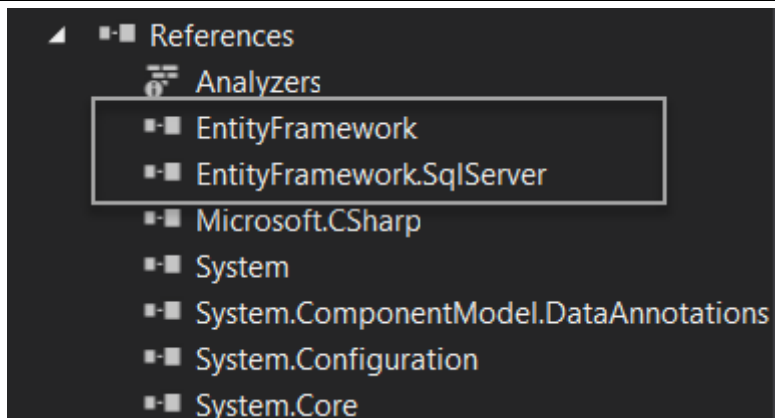
Các Models trong thư mục Models



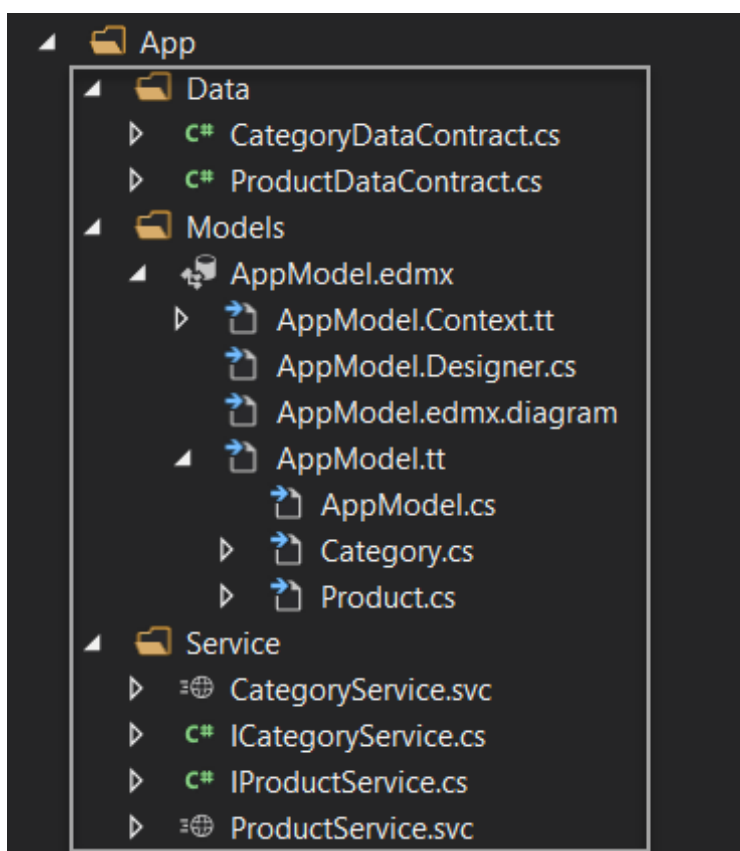
MyConnection là đối tượng DbContext, thông qua đó có thể sử dụng để thao tác với CSDL, Ví dụ

```
private MyConnection _conn = new MyConnection();
```

Entity Framework đã được add vào References



Cấu trúc thư mục sau khi tạo các bước trên là



Bước 8: Mở các interface trong thư mục Service lần lượt khai báo như sau

ICategoryService

```
using System.Collections.Generic;
using System.ServiceModel;
using WcfProductService.App.Data;

namespace WcfProductService.App.Service
{
    [ServiceContract]
    public interface ICategoryService
```

```
{
    [OperationContract]
    List<CategoryDataContract> GetCategorys();

    [OperationContract]
    CategoryDataContract GetCategoryById(int id);

    [OperationContract]
    int CreateNewCategory(string name);

    [OperationContract]
    int UpdateCategory(CategoryDataContract obj);

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

IProductService

```
using System.Collections.Generic;
using System.ServiceModel;
using WcfProductService.App.Data;

namespace WcfProductService.App.Service
{
    [ServiceContract]
    public interface IProductService
    {
        [OperationContract]
        List<ProductDataContract> GetProducts();

        [OperationContract]
        ProductDataContract GetProductById(int id);

        [OperationContract]
        int CreateNewProduct(ProductDataContract obj);

        [OperationContract]
        int UpdateProduct(ProductDataContract obj);

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

Bước 9: Mở các file svc trong thư mục Service lần lượt code như sau

CategoryService.svc triển khai các phương thức đã cài đặt như sau

```
using System.Collections.Generic;
using System.Linq;
using WcfProductService.App.Data;
using WcfProductService.App.Models;

namespace WcfProductService.App.Service
{
    public class CategoryService : ICategoryService
    {
        private MyConnection _conn = new MyConnection();
        public int CreateNewCategory(string name)
        {
            Category category = new Category();
            category.Name = name;
            _conn.Category.Add(category);
            return _conn.SaveChanges();
        }
    }
}
```



```
public int DeleteCategory(int id)
{
    Category category = _conn.Category.Find(id);
    if(category != null)
    {
        _conn.Category.Remove(category);
        return _conn.SaveChanges();
    }
    return 0;
}

public CategoryDataContract GetCategoryById(int id)
{
    CategoryDataContract cat = new CategoryDataContract();
    Category category = _conn.Category.Find(id);
    if (category != null)
    {
        cat.Id = category.Id;
        cat.Name = category.Name;
    }
    return cat;
}

public List<CategoryDataContract> GetCategories()
{
    List<CategoryDataContract> cats = new List<CategoryDataContract>();
    List<Category> categories = _conn.Category.ToList();
    foreach(Category cat in categories)
    {
        cats.Add(new CategoryDataContract
        {
            Id=cat.Id,
            Name = cat.Name
        });
    }

    return cats;
}

public int UpdateCategory(CategoryDataContract obj)
{
    Category category = _conn.Category.Find(obj.Id);
    category.Name = obj.Name;
    _conn.Entry(category).State = System.Data.Entity.EntityState.Modified;
    return _conn.SaveChanges();
}
}
```

ProductService.svc triển khai các phương thức đã cài đặt như sau

```
using System.Collections.Generic;
using System.Linq;
using WcfProductService.App.Data;
using WcfProductService.App.Models;

namespace WcfProductService.App.Service
{
    public class ProductService : IProductService
    {
        private MyConnection _conn = new MyConnection();
        public int CreateNewProduct(ProductDataContract obj)
        {
            Product product = new Product();
            product.Title = obj.Title;
            product.Price = obj.Price;
            product.SalePrice = obj.SalePrice;
            product.Descriptions = obj.Descriptions;
            product.ImageLink = obj.ImageLink;
            product.CategoryId = obj.CategoryId;
        }
    }
}
```



```
_conn.Product.Add(product);  
return _conn.SaveChanges();  
}  
  
public int DeleteProduct(int id)  
{  
    Product product = _conn.Product.Find(id);  
    if (product != null)  
    {  
        _conn.Product.Remove(product);  
        return _conn.SaveChanges();  
    }  
    return 0;  
}  
  
public ProductDataContract GetProductById(int id)  
{  
    ProductDataContract pro = new ProductDataContract();  
    Product product = _conn.Product.Find(id);  
    if (product != null)  
    {  
        pro.Id = product.Id;  
        pro.Title = product.Title;  
        pro.CategoryId = product.CategoryId;  
        pro.Price = product.Price;  
        pro.SalePrice = product.SalePrice;  
        pro.ImageLink = product.ImageLink;  
        pro.Descriptions = product.Descriptions;  
    }  
    return pro;  
}  
  
public List<ProductDataContract> GetProducts()  
{  
    List<ProductDataContract> prods = new List<ProductDataContract>();  
    List<Product> Products = _conn.Product.ToList();  
    foreach (Product pro in Products)  
    {  
        prods.Add(new ProductDataContract  
        {  
            Id = pro.Id,  
            Title = pro.Title,  
            Price = pro.Price,  
            SalePrice = pro.SalePrice,  
            CategoryId = pro.CategoryId,  
            Descriptions = pro.Descriptions,  
            ImageLink = pro.ImageLink  
        });  
    }  
}
```



```
        return prods;
    }

    public int UpdateProduct(ProductDataContract obj)
    {
        Product product = _conn.Product.Find(obj.Id);
        product.Title = obj.Title;
        product.Price = obj.Price;
        product.SalePrice = obj.SalePrice;
        product.Descriptions = obj.Descriptions;
        product.ImageLink = obj.ImageLink;
        product.CategoryId = obj.CategoryId;
        _conn.Entry(product).State = System.Data.Entity.EntityState.Modified;
        return _conn.SaveChanges();
    }
}
```

Bước 9: Test dịch vụ đã tạo trên đây bằng cách

Click phải chuột lên file CategoryService.svc và PrpductService.svc chọn View in Browser

Dự án khởi động trên trình duyệt với link có dạng

<http://localhost:54333/App/Service/CategoryService.svc>

CategoryService 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:54333/App/Service/CategoryService.svc?wsdl
```

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

```
http://localhost:54333/App/Service/CategoryService.svc?singleWsdl
```

<http://localhost:54333/App/Service/ProductService.svc>



ProductService 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:54333/App/Service/ProductService.svc?wsdl
```

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

```
http://localhost:54333/App/Service/ProductService.svc?singleWsdl
```

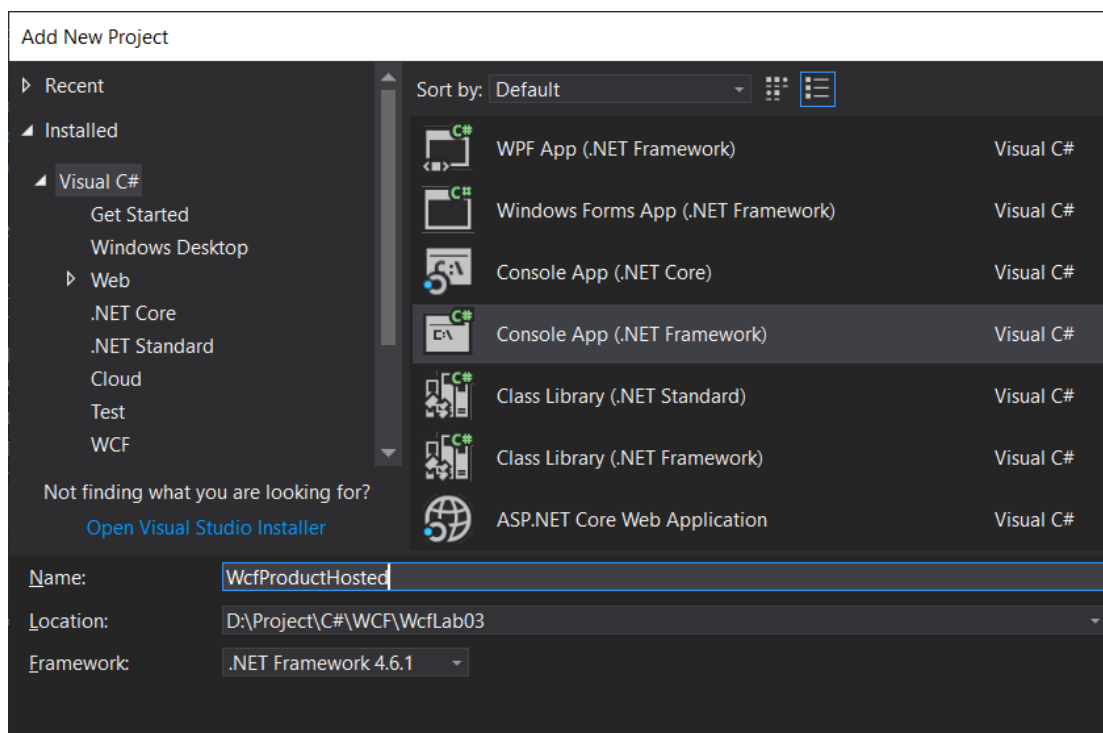
Mở ứng dụng **WCF Test Client** lên và Add Link vào và test như lab02

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

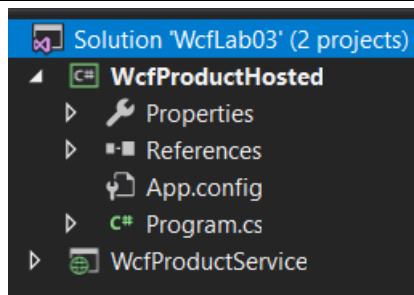
BÀI 2: CẤU HÌNH HOST VÀ ENDPOINT CHO WCF SERVICE

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

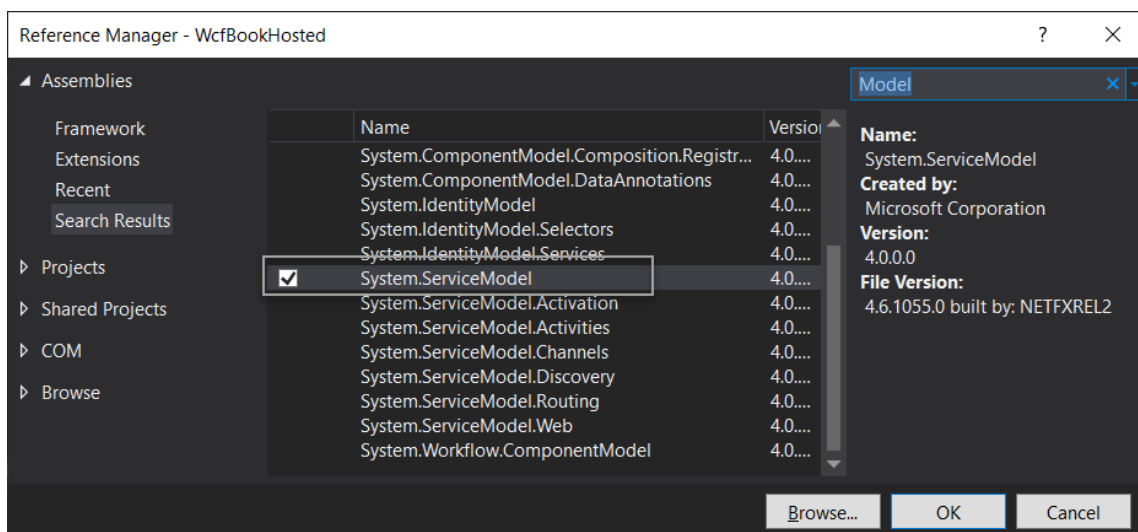


Cấu trúc dự án sau bước trên

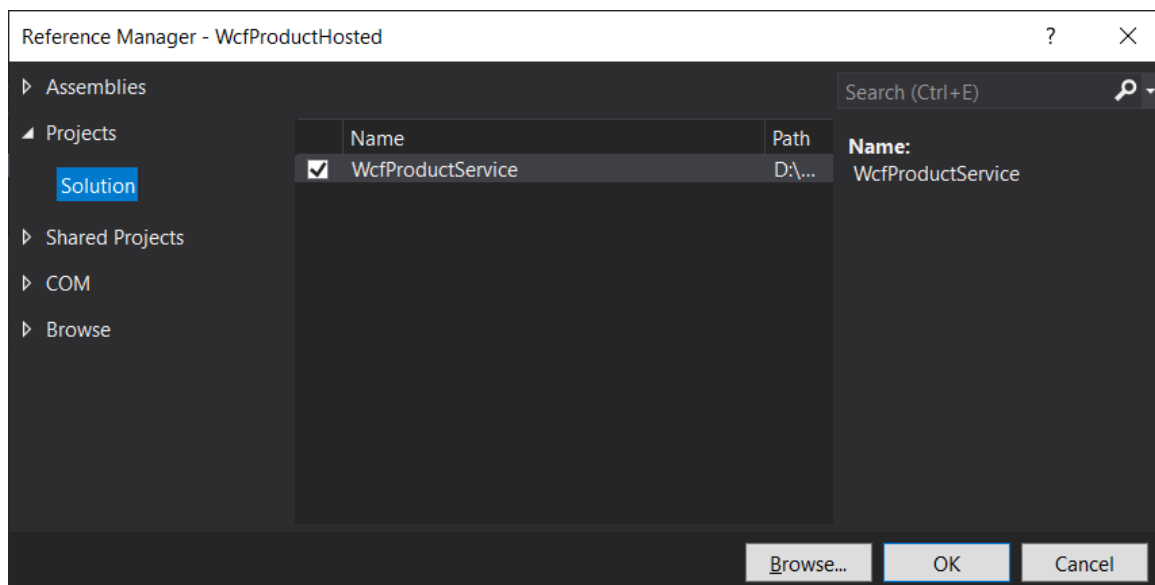


Bước 2: Add các References

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



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



Bước 3: Mở file App.config và copy chuỗi connectionStrings bên dự án WcfProductservice 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;User Id=sa;Password=123a@; Database=WcfProduct" />
    <add name="MyConnection" connectionString="metadata=res://*/App.Models.AppModel.csdl|res://*/App.Mode
ls.AppModel.ssdl|res://*/App.Models.AppModel.msl;provider=System.Data.SqlClient;provider connection string=&q
uot;data source=DESKTOP-
VLQ3DSO\SQL2017;initial catalog=WcfProduct;user id=sa;password=123a@;MultipleActiveResultSets=True;App=Entity
Framework&quot;" providerName="System.Data.EntityClient" />
  </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;User Id=sa;Password=123a@; Database=WcfProduct" />
    <add name="MyConnection" connectionString="metadata=res://*/App.Models.AppModel.csdl|res://*/App.Mode
ls.AppModel.ssdl|res://*/App.Models.AppModel.msl;provider=System.Data.SqlClient;provider connection string=&q
uot;data source=DESKTOP-
VLQ3DSO\SQL2017;initial catalog=WcfProduct;user id=sa;password=123a@;MultipleActiveResultSets=True;App=Entity
Framework&quot;" providerName="System.Data.EntityClient" />
  </connectionStrings>
  <system.serviceModel>
    <behaviors>
      <serviceBehaviors>
        <behavior name="MyBehavior">
          <serviceMetadata httpGetEnabled="true"/>
        </behavior>
      </serviceBehaviors>
    </behaviors>
    <services>
      <service name="WcfProductService.App.Service.CategoryService" behaviorConfiguration="MyBehavior">
        <host>
          <baseAddresses>
            <add baseAddress="http://localhost:33/category"/>
          </baseAddresses>
        </host>
        <endpoint address="CategoryService" contract="WcfProductService.App.Service.ICategoryService"
binding="basicHttpBinding" />
      </service>
      <service name="WcfProductService.App.Service.ProductService" behaviorConfiguration="MyBehavior">
        <host>
          <baseAddresses>
            <add baseAddress="http://localhost:33/product"/>
          </baseAddresses>
        </host>
        <endpoint address="ProductService" contract="WcfProductService.App.Service.IProductService" b
inding="basicHttpBinding" />
      </service>
    </services>
  </system.serviceModel>
</configuration>
```

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



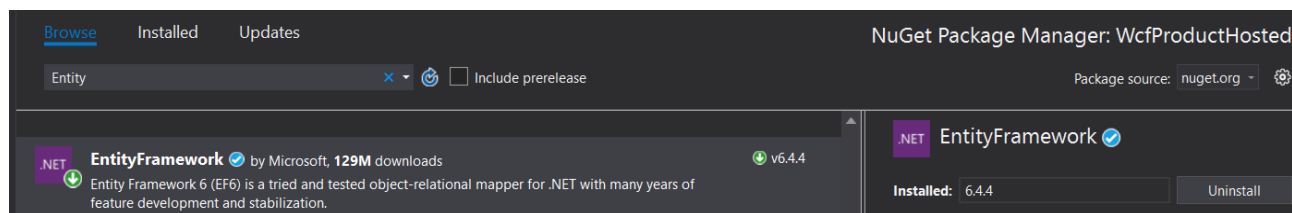
Như code trên đây có thể thấy có 2 endpoint

<http://localhost:33/category>

<http://localhost:33/product>

Bước 5: Cài đặt Entity Framework cho dự án **WcfHosted** này

Mở Manager NuGET Packages... lên tìm đến EntityFramework và tiến hành cài đặt



Bước 6: Mở file Program.cs lên và soạn code mở host như sau

```
using System;
using System;
using System.ServiceModel;
using System.Text;

namespace WcfProductHosted
{
    class Program
    {
        static void Main(string[] args)
        {
            ServiceHost hostA = new ServiceHost(typeof(WcfProductService.App.Service.CategoryService));
            ServiceHost hostB = new ServiceHost(typeof(WcfProductService.App.Service.ProductService));
            hostA.Open();
            hostB.Open();
            Console.OutputEncoding = Encoding.UTF8; // hiển thị tiếng việt trên màn hình
            Console.WriteLine("-> MỞ CỘNG DỊCH VỤ ----->");
            Console.WriteLine("-> Category service http://localhost:33/category");
            Console.WriteLine("-> Product service http://localhost:33/product");
            Console.ReadLine();
        }
    }
}
```

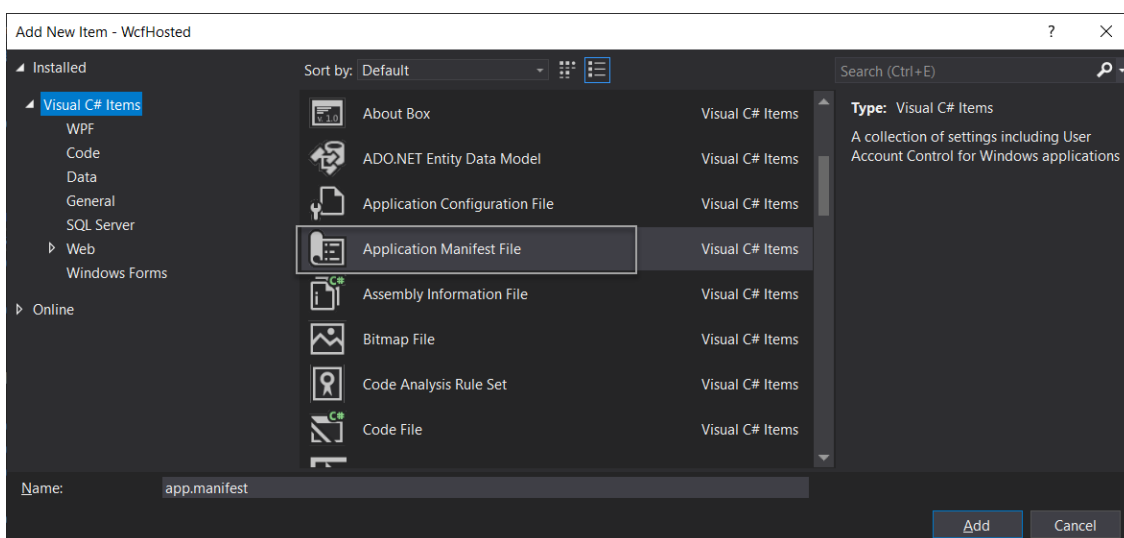
Build lại dự án chạy sẽ có lỗi tương tự như sau



```
C:\Windows\system32\cmd.exe

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 . . .
```

Lỗi khi khởi động

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 **WcfProductHosted** và chọn Add -> new Item và chọn đến **Application Manifest File**, để nguyên tên file là app.manifestSau đó tìm và cấu hình thuộc tính **requestedExecutionLevel**

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

```
Select D:\Project\C#\WCF\WcfLab03\WcfProductHosted\bin\Debug\
-> MỞ CỘNG DỊCH VỤ ----->
-> Category service http://localhost:33/category
-> Product service http://localhost:33/product
```

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

<http://localhost:33/category>

CategoryService 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:

<http://localhost:33/product>

ProductService Service

You have created a service.

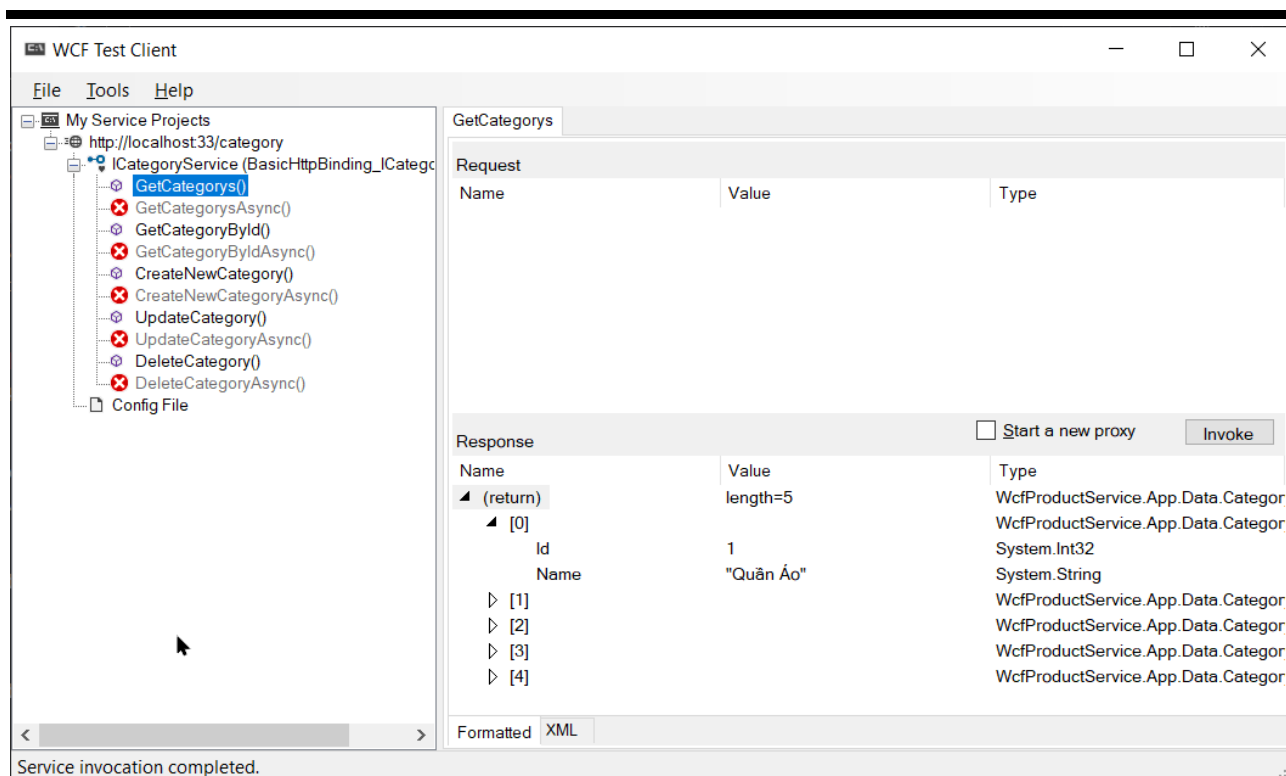
To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://localhost:33/product?wsdl
```

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

```
http://localhost:33/product?singleWsd1
```

Chú ý: Hãy tiến hành sử dụng **WCF Test Client** để test các phương thức của CategoryService và ProductService để đảm bảo các phương thức hoạt động đúng VD, phương thức GetCategorys()



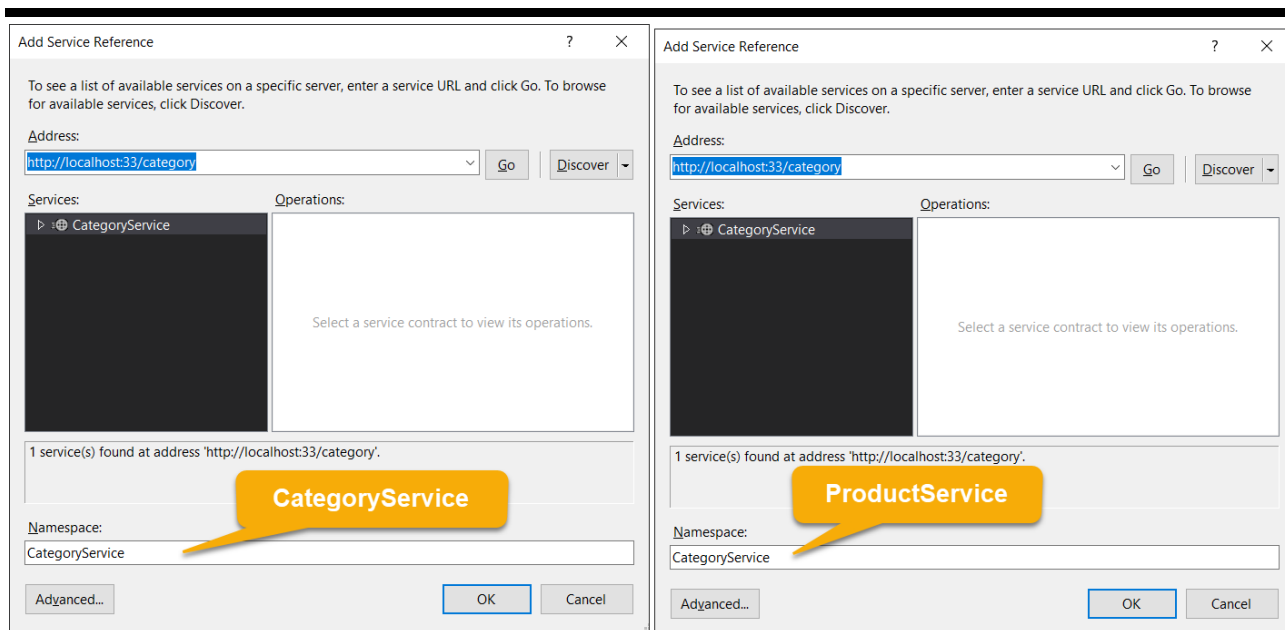
Các phương thức khác tương tự như bài lab02

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

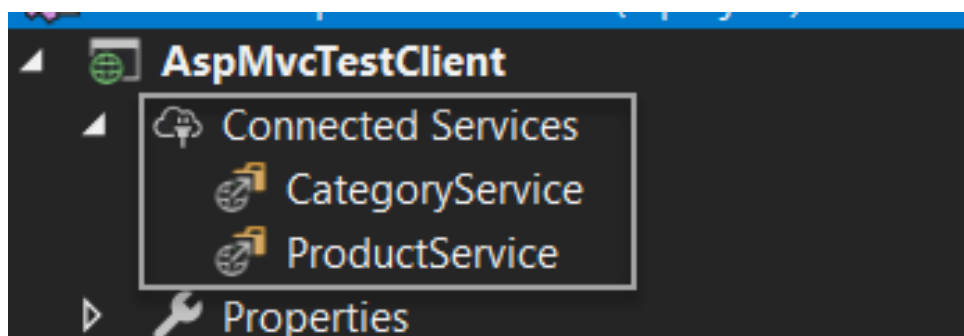
Bước 1: Mở thêm Visual Studio 2017 lên tạo dự án .NET MVC mới đặt tên là AspMvcTestClient

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

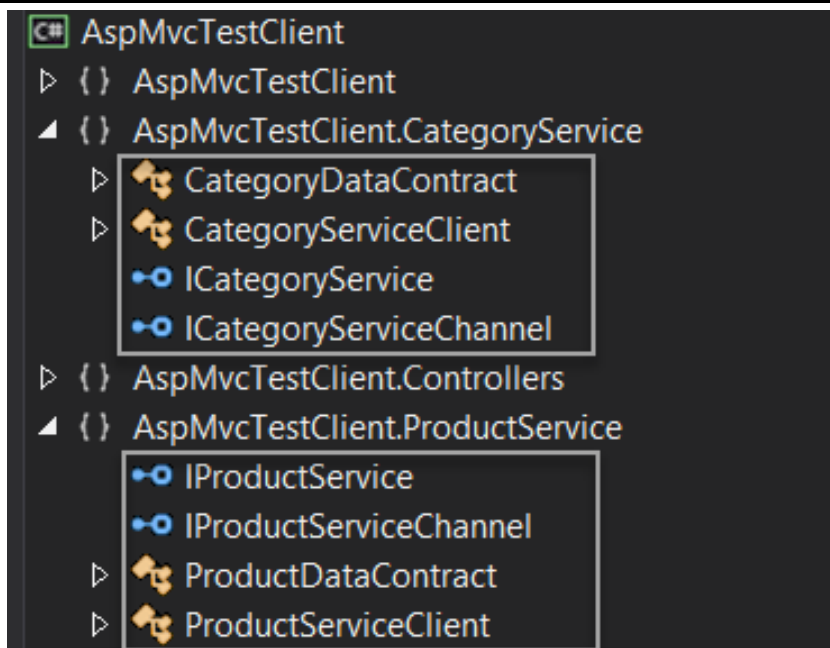
Sau khi màn hình sau đây sẽ hiển thị hãy nhập url của 2 dịch vụ đã khởi động ở trên
Theo hình sau



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



Click vào namespace CategoryService hoặc ProductService sẽ thấy các thành phần được sinh ra tương ứng như sau



Bước 3: Tạo một Controller mới đặt tên là ProductController

Tại ActionResult Index code C# sử dụng ProductService như sau

```
using AspMvcTestClient.ProductService;
using System.Web.Mvc;

namespace AspMvcTestClient.Controllers
{
    public class ProductController : Controller
    {
        // Khởi tạo đối tượng ProductServiceClient
        private ProductServiceClient client = new ProductServiceClient();
        public ActionResult Index()
        {
            // Phương thức GetProducts để lấy về danh sách
            ViewBag.products = client.GetProducts();
            client.Close(); // đóng dịch vụ
            return View();
        }
    }
}
```

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

```
@{
    ViewBag.Title = "Product Index";
    var products = ViewBag.products;
}
<h2>Danh sách sản phẩm</h2>
<hr />
<form class="form-inline" >
    <div class="form-group">
        <input name="key" class="form-control" placeholder="Tìm kiếm theo tên..."/>
    </div>
```

```
<button class="btn btn-warning">Tìm kiếm</button>
<a href="@Url.Action("Create","Product")" class="btn btn-primary">Thêm mới</a>
</form>
<hr />
<table class="table table-bordered">
  <thead>
    <tr>
      <th>Id</th>
      <th>Name</th>
      <th>Price</th>
      <th>Sale Price</th>
      <th>Category Id</th>
      <th>Image</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    @foreach (var item in products)
    {
      <tr>
        <td>@item.Id</td>
        <td>@item.Title</td>
        <td>@item.Price</td>
        <td>@item.SalePrice</td>
        <td>@item.CategoryId</td>
        <td></td>
        <td>
          <a href="@Url.Action("Detail","Product", new { id = item.Id })" class="btn btn-sm btn-
primary">Chi tiết</a>
          <a href="@Url.Action("Edit","Product", new { id = item.Id })" class="btn btn-sm btn-
success">Chi tiết</a>
          <a href="@Url.Action("Delete","Product", new { id = item.Id })" class="btn btn-sm btn-
danger" onclick="return confirm('Bạn có chắc muốn xóa?')">Chi tiết</a>
        </td>
      </tr>
    }
  </tbody>
</table>
```

Build lại dự án này và truy cập link xem kết quả <http://localhost:56816/Product>



Danh sách sản phẩm

Tìm kiếm theo tên...

Tìm kiếm

Thêm mới

Id	Name	Price	Sale Price	Category Id	Image	
1	Áo sơ mi nam	500000	4500	1		Chi tiết Chỉnh sửa Xóa bỏ
2	Demo Sản phẩm 2	500000	250	1		Chi tiết Chỉnh sửa Xóa bỏ
3	Demo Sản phẩm 3	500000	350000	1		Chi tiết Chỉnh sửa Xóa bỏ

Bước 5: Để tạo chức năng thêm mới, mở lại ProductController 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 AspMvcTestClient.ProductService;
using AspMvcTestClient.CategoryService;
using System.Web.Mvc;
using System;

namespace AspMvcTestClient.Controllers
{
    public class ProductController : Controller
    {
        private ProductServiceClient client = new ProductServiceClient();
        private CategoryServiceClient catclient = new CategoryServiceClient();
        public ActionResult Index()
        {
            // Phương thức GetProducts để lấy về danh sách
            ViewBag.products = client.GetProducts();
            client.Close(); // đóng dịch vụ
            return View();
        }
        public ActionResult Create()
        {
            // gửi dữ liệu danh mục qua form thêm mới sản phẩm
            ViewBag.cats = catclient.GetCategories();
            return View();
        }
        [HttpPost]
        public ActionResult Create(FormCollection collection)
        {
            ProductDataContract product = new ProductDataContract();
            product.Title = collection["Title"];
            product.Price = Convert.ToDouble(collection["Price"]);
            product.SalePrice = Convert.ToDouble(collection["SalePrice"]);
            product.CategoryId = Convert.ToInt32(collection["CategoryId"]);
            product.Descriptions = collection["Descriptions"];
            product.ImageLink = collection["ImageLink"];
            // Lấy dữ liệu trên form và gọi dịch vụ thêm mới
            int isCreated = client.CreateNewProduct(product);
            if(isCreated == 1)
            {
                return RedirectToAction("Index", "Product");
            }
            else
            {
                return RedirectToAction("Create", "Product");
            }
        }
    }
}
```




```
}  
  
}  
}
```

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

```
@{  
    ViewBag.Title = "Create";  
}  
  
<h2>Thêm mới sản phẩm</h2>  
<form action="" method="post">  
    <div class="row">  
        <div class="col-md-4">  
            <div class="form-group">  
                <label>Danh mục sản phẩm</label>  
  
                <select name="CategoryId" class="form-control">  
                    <option value="">Chọn danh mục</option>  
                    @foreach (var cat in ViewBag.cats)  
                    {  
                        <option value="@cat.Id">@cat.Name</option>  
                    }  
                </select>  
  
            </div>  
            <div class="form-group">  
                <label>Tên sản phẩm</label>  
                <input name="Title" class="form-control" placeholder="Tên sản phẩm ..." />  
            </div>  
            <div class="form-group">  
                <label>Giá sản phẩm</label>  
                <input name="Price" class="form-control" placeholder="Giá sản phẩm ..." />  
            </div>  
            <div class="form-group">  
                <label>Giá khuyến mãi</label>  
                <input name="SalePrice" class="form-control" placeholder="Giá khuyến mãi ..." />  
            </div>  
            <div class="form-group">  
                <label>Link ảnh</label>  
                <input name="ImageLink" class="form-control" placeholder="Link ảnh ..." />  
            </div>  
            <div class="form-group">  
                <label>Mô tả</label>  
                <textarea name="Descriptions" class="form-control" placeholder="Mô tả sản phẩm ..." />  
            </div>  
            <button class="btn btn-sm btn-primary">Lưu lại</button>  
            <a href="@Url.Action("Index", "Product")" class="btn btn-sm btn-default">Quay lại</a>  
        </div>  
    </div>  
</form>
```

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

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



Thêm mới sản phẩm

Danh mục sản phẩm

Tên sản phẩm

Giá sản phẩm

Giá khuyến mãi

Link ảnh

Mô tả

Hãy nhập dữ liệu và test thử thêm mới

Bước 7: Trong ProductController 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 AspMvcTestClient.ProductService;
using AspMvcTestClient.CategoryService;
using System.Web.Mvc;
using System;

namespace AspMvcTestClient.Controllers
{
    public class ProductController : Controller
    {
        // Khởi tạo đối tượng ProductServiceClient
        private ProductServiceClient client = new ProductServiceClient();
        private CategoryServiceClient catclient = new CategoryServiceClient();
        // các phương thức khác vẫn để ở đây
        public ActionResult Edit(int id)
        {
            ViewBag.cats = catclient.GetCategorys();
            ViewBag.product = client.GetProductById(id);
        }
    }
}
```



```
        return View();
    }

    [HttpPost]
    public ActionResult Edit(int id, FormCollection collection)
    {
        ProductDataContract product = new ProductDataContract();
        product.Id = id;
        product.Title = collection["Title"];
        product.Price = Convert.ToDouble(collection["Price"]);
        product.SalePrice = Convert.ToDouble(collection["SalePrice"]);
        product.CategoryId = Convert.ToInt32(collection["CategoryId"]);
        product.Descriptions = collection["Descriptions"];
        product.ImageLink = collection["ImageLink"];

        int isUpdated = client.UpdateProduct(product);
        if (isUpdated == 1)
        {
            return RedirectToAction("Index", "Product");
        }
        else
        {
            return RedirectToAction("Edit", "Product");
        }
    }
}
```

Bước 8: Add View Edit có code như sau

```
@{
    ViewBag.Title = "Product Edit";
    var product = ViewBag.product;
}
<h2>Chỉnh sửa sản phẩm</h2>
<form action="" method="post">
    <div class="row">
        <div class="col-md-4">
            <div class="form-group">
                <label>Danh mục sản phẩm</label>

                <select name="CategoryId" class="form-control">
                    <option value="">Chọn danh mục</option>
                    @foreach (var cat in ViewBag.cats)
                    {
                        if (cat.Id == product.CategoryId)
                        {
                            <option value="@cat.Id" selected>@cat.Name</option>
                        }
                        else
                        {
                            <option value="@cat.Id">@cat.Name</option>
                        }
                    }
                </select>
            </div>
            <div class="form-group">
                <label>Tên sản phẩm</label>
                <input name="Title" class="form-control" value="@product.Title" />
            </div>
            <div class="form-group">
                <label>Giá sản phẩm</label>
                <input name="Price" class="form-control" value="@product.Price" />
            </div>
            <div class="form-group">
                <label>Giá khuyến mãi</label>
                <input name="SalePrice" class="form-control" value="@product.SalePrice" />
            </div>
            <div class="form-group">
                <label>Link ảnh</label>
                <input name="ImageLink" class="form-control" value="@product.ImageLink" />
            </div>
        </div>
    </div>
</form>
```



```
</div>
<div class="form-group">
  <label>Mô tả </label>
  <textarea name="Descriptions" class="form-
control" placeholder="Nội dung ...">@product.Descriptions</textarea>
</div>
<button class="btn btn-sm btn-primary">Lưu lại</button>
<a href="@Url.Action("Index","Product")" class="btn btn-sm btn-default">Quay lại</a>
</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 sản phẩm

Danh mục sản phẩm

Quần Áo

Tên sản phẩm

Áo sơ mi nam

Giá sản phẩm

500000

Giá khuyến mãi

4500

Link ảnh

https://baoquoc.vn/stores/news_dataim

Mô tả

Demo ta áo sơ mi nam

Lưu lại

Quay lại

Bước 9: Thêm Action Detail

Code C#

```
using AspMvcTestClient.ProductService;
using AspMvcTestClient.CategoryService;
using System.Web.Mvc;
using System;

namespace AspMvcTestClient.Controllers
{
    public class ProductController : Controller
    {
        // Khởi tạo đối tượng ProductServiceClient
        private ProductServiceClient client = new ProductServiceClient();
        private CategoryServiceClient catclient = new CategoryServiceClient();
        // các phương thức khác vẫn để đây
        public ActionResult Detail(int id)
        {
            ViewBag.product = client.GetProductById(id);
            return View();
        }
    }
}
```



Code trên view

```
@{
    ViewBag.Title = "Product Detail";
    var product = ViewBag.product;
}

<h2>Chi tiết sản phẩm</h2>
<div class="row">
    <div class="col-md-5">
        
    </div>
    <div class="col-md-7">
        <h2>@product.Title</h2>
        <h4>Giá: @product.Price</h4>
        <h4>Giá KM: @product.SalePrice</h4>
        <p>
            @product.Descriptions
        </p>
    </div>
</div>
<hr />
<a href="@Url.Action("Index", "Product")" class="btn btn-sm btn-primary">Về danh sách</a>
<a href="@Url.Action("Edit", "Product", new { id = product.Id})" class="btn btn-sm btn-success">Sửa</a>
<a href="@Url.Action("Delete", "Product", new { id = product.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:49981/Product/Detail/1>

Chi tiết sản phẩm



Áo sơ mi nam

Giá: 500000

Giá KM: 4500

Demo ta áo sơ mi nam

Về danh sách

Sửa

Xóa

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

```
using AspMvcTestClient.ProductService;
using AspMvcTestClient.CategoryService;
using System.Web.Mvc;
using System;

namespace AspMvcTestClient.Controllers
{
    public class ProductController : Controller
    {
        // Khởi tạo đối tượng ProductServiceClient
        private ProductServiceClient client = new ProductServiceClient();
        private CategoryServiceClient catclient = new CategoryServiceClient();
        // các phương thức khác vẫn để ở đây
        public ActionResult Delete(int id)
        {
            int isDeleted = client.DeleteProduct(id);
            return RedirectToAction("Index", "Product");
        }
    }
}
```

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

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

Toàn bộ code của ProductController

```
using AspMvcTestClient.ProductService;
using AspMvcTestClient.CategoryService;
using System.Web.Mvc;
using System;

namespace AspMvcTestClient.Controllers
{
    public class ProductController : Controller
    {
        // Khởi tạo đối tượng ProductServiceClient
        private ProductServiceClient client = new ProductServiceClient();
        private CategoryServiceClient catclient = new CategoryServiceClient();
        public ActionResult Index()
        {
            // Phương thức GetProducts để lấy về danh sách
            ViewBag.products = client.GetProducts();
            client.Close(); // đóng dịch vụ
            return View();
        }

        public ActionResult Create()
        {
            ViewBag.cats = catclient.GetCategories();
            return View();
        }

        [HttpPost]
        public ActionResult Create(FormCollection collection)
        {
            ProductDataContract product = new ProductDataContract();
            product.Title = collection["Title"];
            product.Price = Convert.ToDouble(collection["Price"]);
            product.SalePrice = Convert.ToDouble(collection["SalePrice"]);
            product.CategoryId = Convert.ToInt32(collection["CategoryId"]);
            product.Descriptions = collection["Descriptions"];
            product.ImageLink = collection["ImageLink"];
        }
    }
}
```



```
int isCreated = client.CreateNewProduct(product);
if(isCreated == 1)
{
    return RedirectToAction("Index", "Product");
}
else
{
    return RedirectToAction("Create", "Product");
}
}

public ActionResult Detail(int id)
{
    ViewBag.product = client.GetProductById(id);
    return View();
}

public ActionResult Edit(int id)
{
    ViewBag.cats = catclient.GetCategorys();
    ViewBag.product = client.GetProductById(id);
    return View();
}

[HttpPost]
public ActionResult Edit(int id, FormCollection collection)
{
    ProductDataContract product = new ProductDataContract();
    product.Id = id;
    product.Title = collection["Title"];
    product.Price = Convert.ToDouble(collection["Price"]);
    product.SalePrice = Convert.ToDouble(collection["SalePrice"]);
    product.CategoryId = Convert.ToInt32(collection["CategoryId"]);
    product.Descriptions = collection["Descriptions"];
    product.ImageLink = collection["ImageLink"];

    int isUpdated = client.UpdateProduct(product);
    if (isUpdated == 1)
    {
        return RedirectToAction("Index", "Product");
    }
    else
    {
        return RedirectToAction("Edit", "Product");
    }
}

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

BÀI TẬP TỰ LÀM

Sinh viên tạo dự án WCF Service các chức năng CRUD cho các bảng sau đây

```
create database WcfTravel
go
use WcfTravel
go
-- Các danh mục thể loại du lịch
Create Table TravelCategory
(
    Id int Primary Key Identity,
    Name nvarchar(100) NOT NULL,
    Status tinyint DEFAULT(1)
)
```




```
GO
-- Giới thiệu các điểm du lịch
Create Table Travel
(
    Id int Primary key Identity,
    Title nvarchar(120) NOT NULL,
    CategoryId int NOT NULL FOREIGN KEY REFERENCES Category(Id),
    Descriptions nvarchar(max) null,
    ImageLink varchar(200) NULL,
)
GO
-- thêm dữ liệu cho thể loại
INSERT INTO TravelCategory (Name) VALUES
(N'Du lịch trong nước'),
(N'Du lịch nước ngoài'),
(N'Du lịch 4 mùa')
```

Sau đó có thể tạo các dự án client như: Winform, Webform, ASP.NET MVC.. để test các dịch vụ trên

