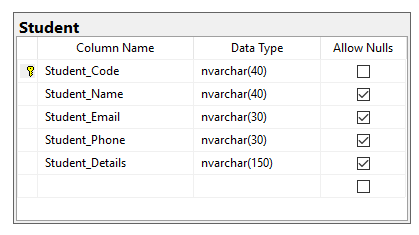
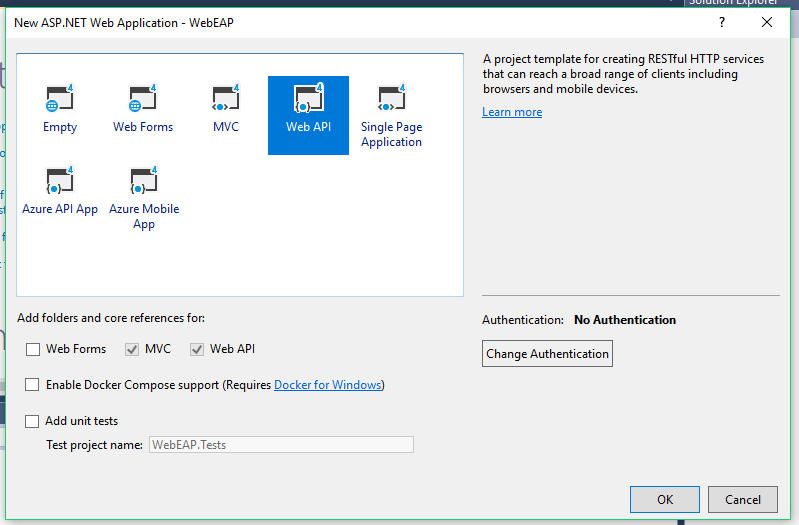
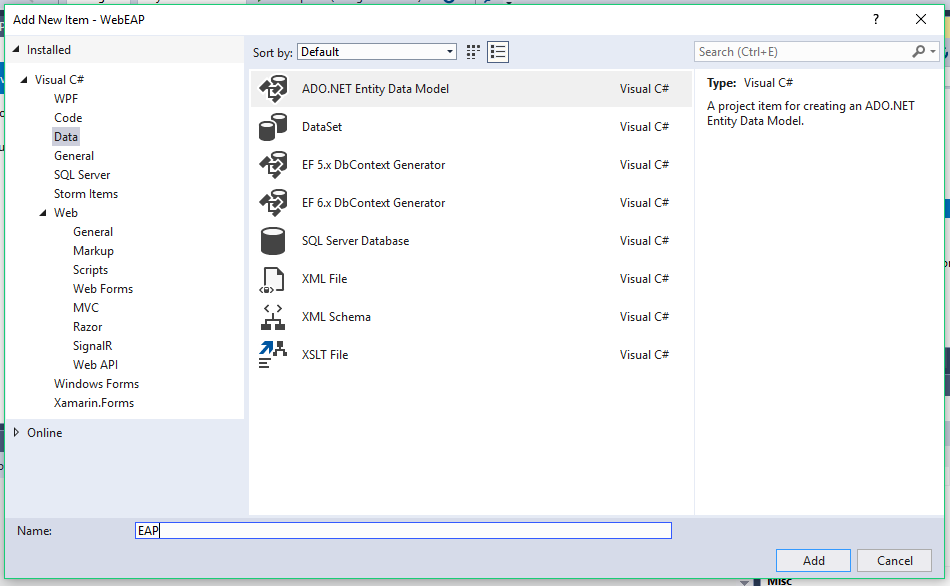
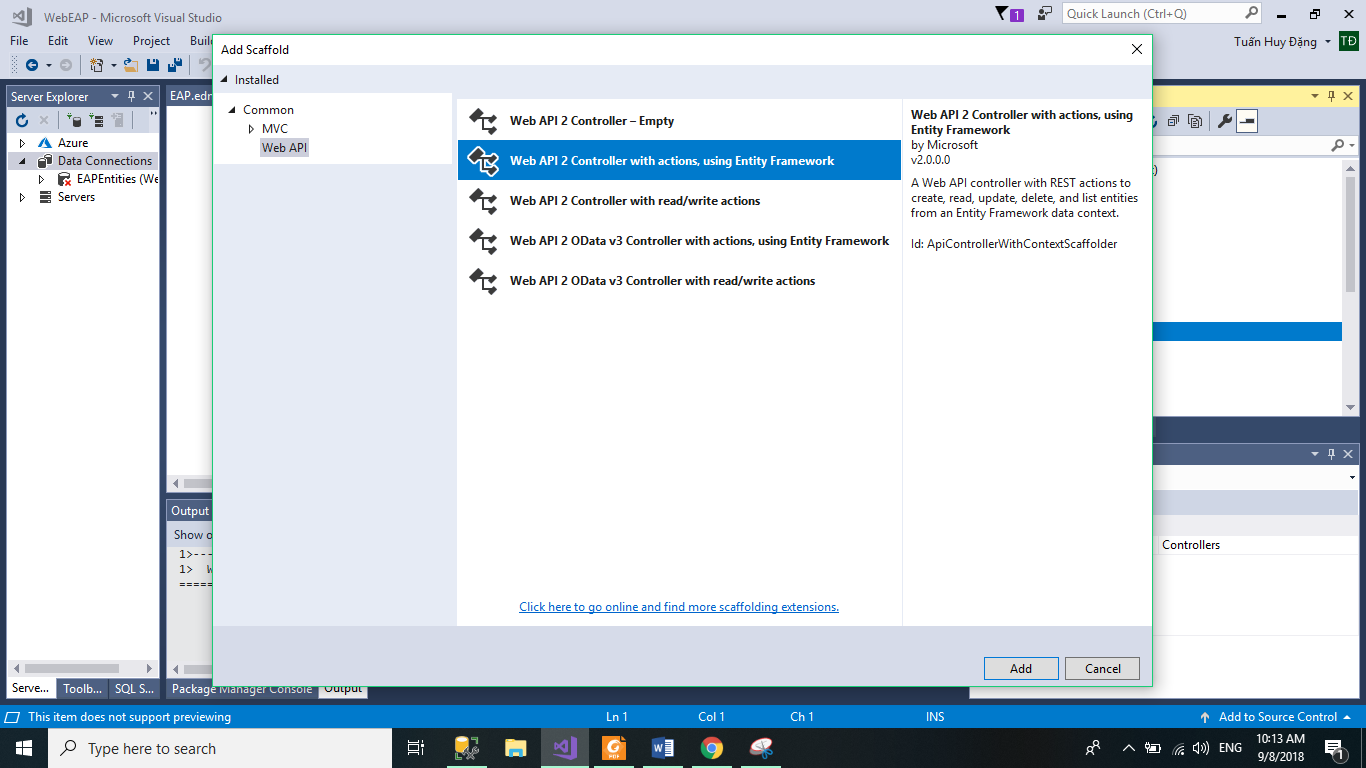
**Reviews EAPs**

1. **CSDL (EAP) – tables Student**
2. **Tạo một Project API-Web**
3. **Sử dụng EntityFramwwork tạo Models**



1. **Tạo controller**



1. **Hiệu chỉnh**

**Trong Application\_Start() bổ sung code convert xml->json**

using System.Web.Http;

using System.Web.Mvc;

using System.Web.Optimization;

using System.Web.Routing;

namespace WebEAP

{

public class WebApiApplication : System.Web.HttpApplication

{

protected void Application\_Start()

{

GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings.ReferenceLoopHandling = Newtonsoft.Json.ReferenceLoopHandling.Ignore;

GlobalConfiguration.Configuration.Formatters.Remove(GlobalConfiguration.Configuration.Formatters.XmlFormatter);

AreaRegistration.RegisterAllAreas();

GlobalConfiguration.Configure(WebApiConfig.Register);

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

RouteConfig.RegisterRoutes(RouteTable.Routes);

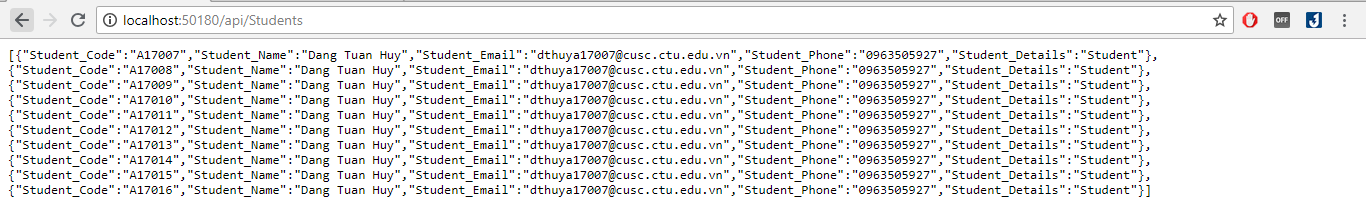
BundleConfig.RegisterBundles(BundleTable.Bundles);

}

}

}

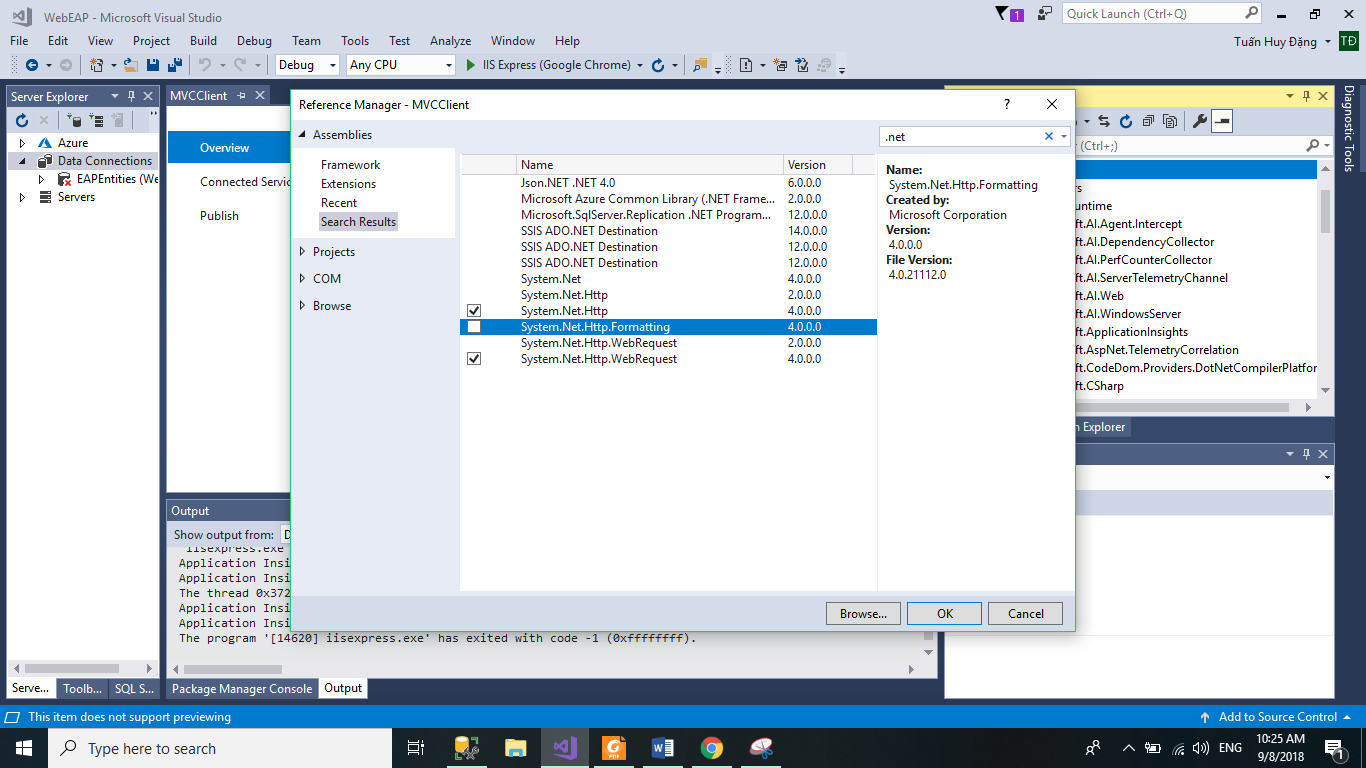
**Kết quả**



1. **API – Client**
2. **Web MVC – Client**

***Trong cùng Solution tạo một Project MVCClient***

***Trong References của MVCClient hiệu chỉnh***

1. **Trong model tạo một Class có tên là Student.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace MVCClient.Models

{

public class Student

{

public string Student\_Code { get; set; }

public string Student\_Name { get; set; }

public string Student\_Email { get; set; }

public string Student\_Phone { get; set; }

public string Student\_Details { get; set; }

}

}

1. **Tạo một controller emty có tên là StudentsController.cs Bổ sung code gọi API**

using MVCClient.Models;

using Newtonsoft.Json;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Threading.Tasks;

using System.Web;

using System.Web.Mvc;

namespace MVCClient.Controllers

{

public class StudentsController : Controller

{

HttpClient client;

string url = "http://localhost:50180/api/Students";

public StudentsController()

{

client = new HttpClient();

client.BaseAddress = new Uri(url);

client.DefaultRequestHeaders.Accept.Clear();

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

}

// Get Student

public async Task<ActionResult> Index()

{

HttpResponseMessage responseMessage = await client.GetAsync(url);

if(responseMessage.IsSuccessStatusCode)

{

var responseData = responseMessage.Content.ReadAsStringAsync().Result;

var students = JsonConvert.DeserializeObject<List<Student>>(responseData);

return View(students);

}

return View("Error");

}

//Form Create

public ActionResult Create()

{

return View(new Student());

}

[HttpPost]

public async Task<ActionResult> Create(Student student)

{

HttpResponseMessage responseMessage = await client.PostAsJsonAsync(url,student);

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return RedirectToAction("Error");

}

//Form Edit

public async Task<ActionResult> Edit(string id)

{

HttpResponseMessage responseMessage = await client.GetAsync(url + "/" + id);

if (responseMessage.IsSuccessStatusCode)

{

var responseData = responseMessage.Content.ReadAsStringAsync().Result;

var student = JsonConvert.DeserializeObject<Student>(responseData);

return View(student);

}

return View("Error");

}

[HttpPost]

public async Task<ActionResult> Edit(string id, Student student)

{

HttpResponseMessage responseMessage = await client.PutAsJsonAsync(url + "/" + id, student);

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return RedirectToAction("Error");

}

//Form Delete

public async Task<ActionResult> Delete(string id)

{

HttpResponseMessage responseMessage = await client.GetAsync(url + "/" + id);

if (responseMessage.IsSuccessStatusCode)

{

var responseData = responseMessage.Content.ReadAsStringAsync().Result;

var student = JsonConvert.DeserializeObject<Student>(responseData);

return View(student);

}

return View("Error");

}

[HttpPost]

public async Task<ActionResult> Delete(string id, Student student)

{

HttpResponseMessage responseMessage = await client.DeleteAsync(url + "/" +id);

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return RedirectToAction("Error");

}

}

}

1. **Tạo Views từ model Student**

**Code Index**

@model IEnumerable<MVCClient.Models.Student>

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<h2>List Student</h2>

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table table-bordered">

<tr>

<th>

@Html.DisplayNameFor(model => model.Student\_Code)

</th>

<th>

@Html.DisplayNameFor(model => model.Student\_Name)

</th>

<th>

@Html.DisplayNameFor(model => model.Student\_Email)

</th>

<th>

@Html.DisplayNameFor(model => model.Student\_Phone)

</th>

<th>

@Html.DisplayNameFor(model => model.Student\_Details)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Student\_Code)

</td>

<td>

@Html.DisplayFor(modelItem => item.Student\_Name)

</td>

<td>

@Html.DisplayFor(modelItem => item.Student\_Email)

</td>

<td>

@Html.DisplayFor(modelItem => item.Student\_Phone)

</td>

<td>

@Html.DisplayFor(modelItem => item.Student\_Details)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id=item.Student\_Code}) |

@Html.ActionLink("Delete", "Delete", new {id = item.Student\_Code })

</td>

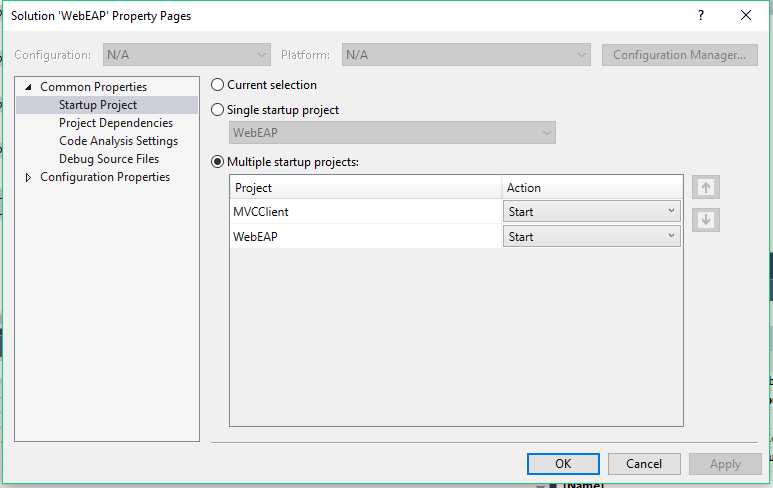
</tr>

}

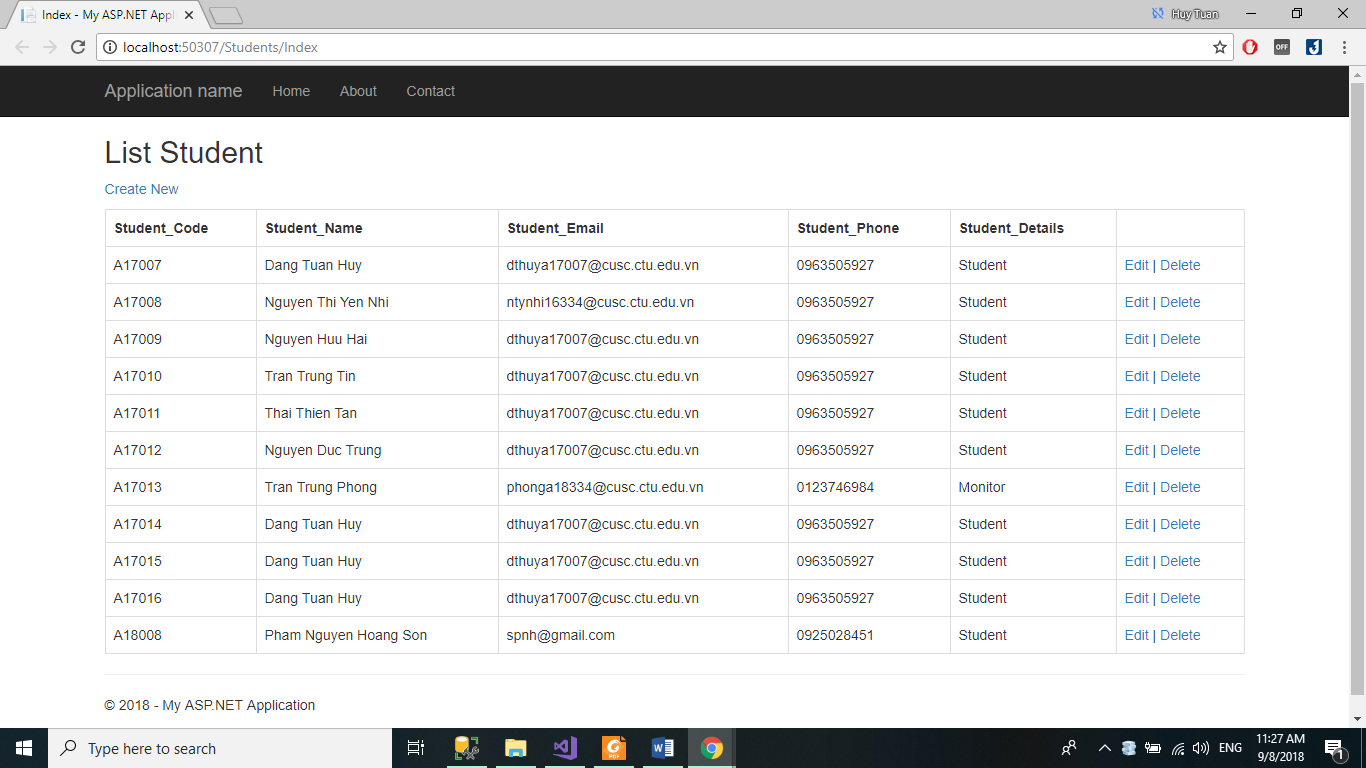
</table>

1. **Kết quả**

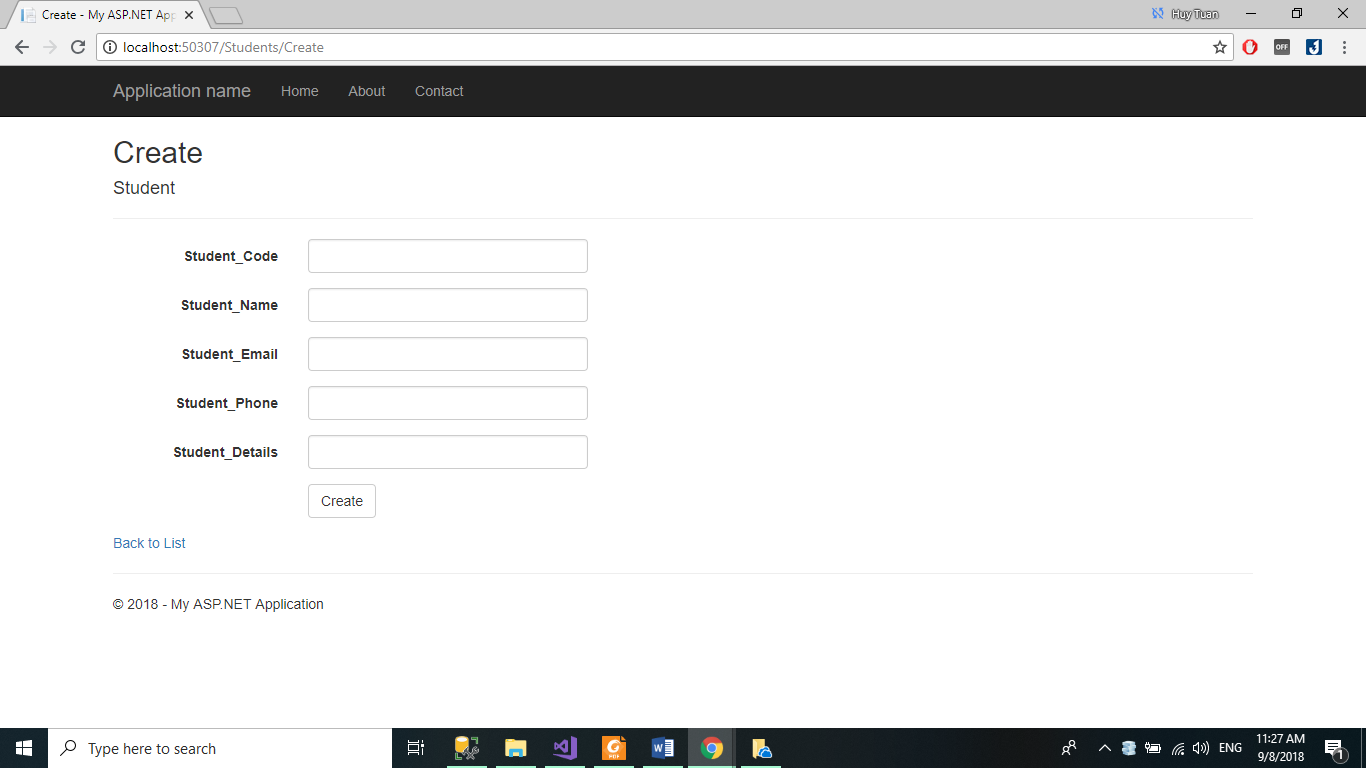
**Lưu ý: config để API chạy cùng với MVC**



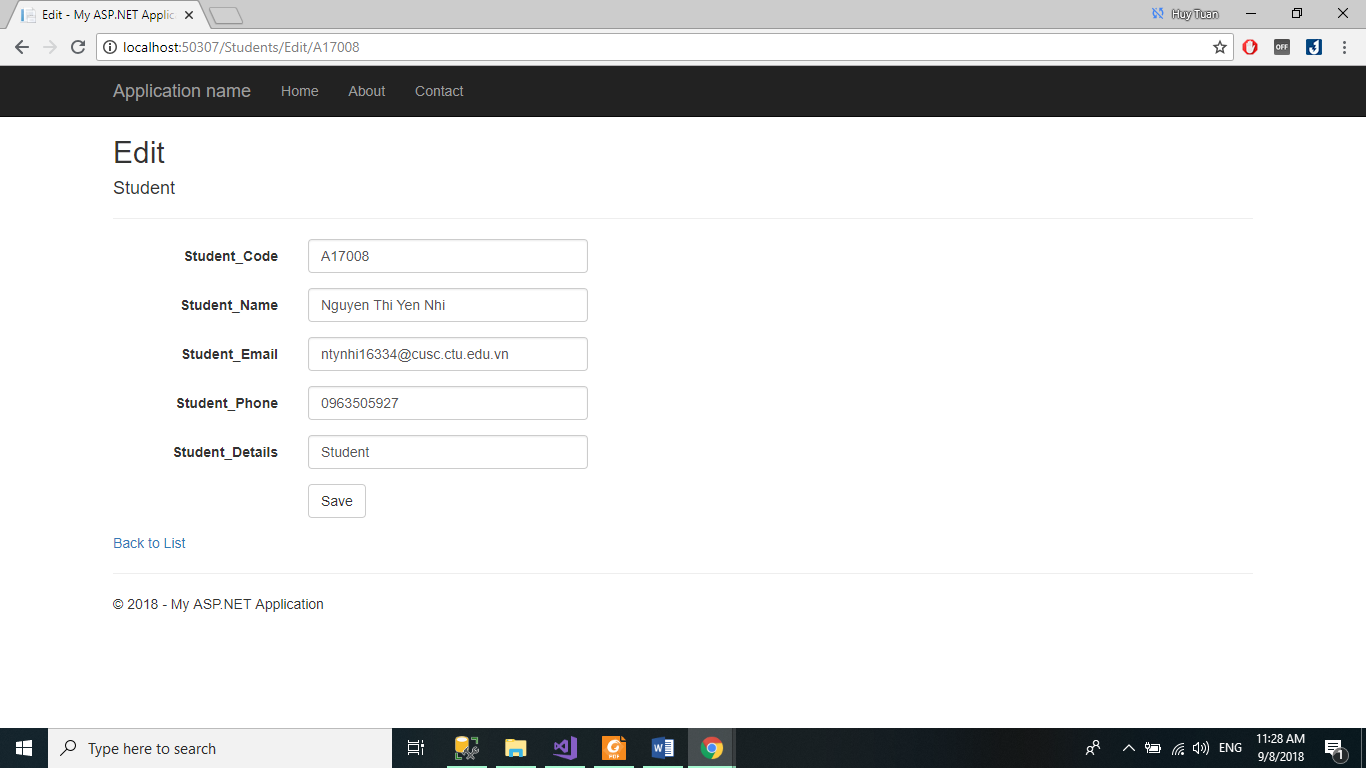
* **Index:**



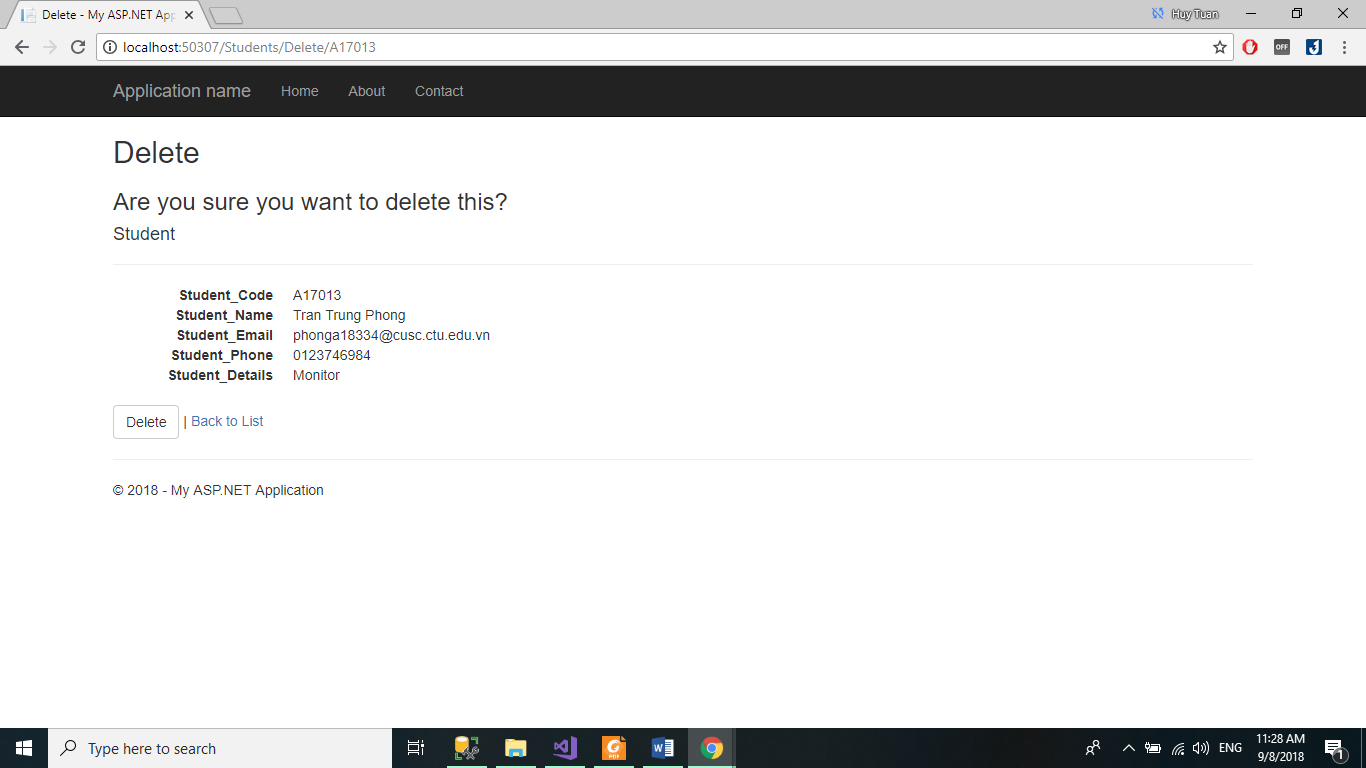
* **Create**

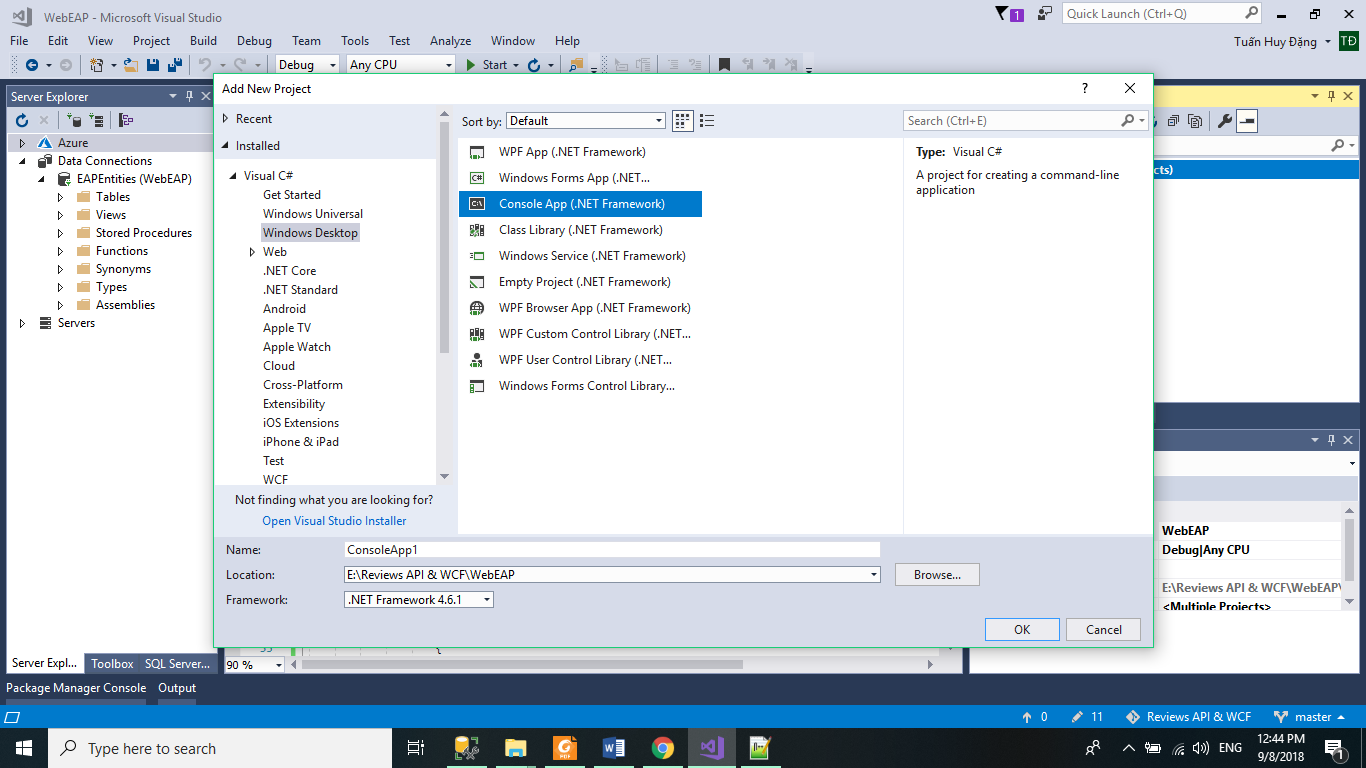
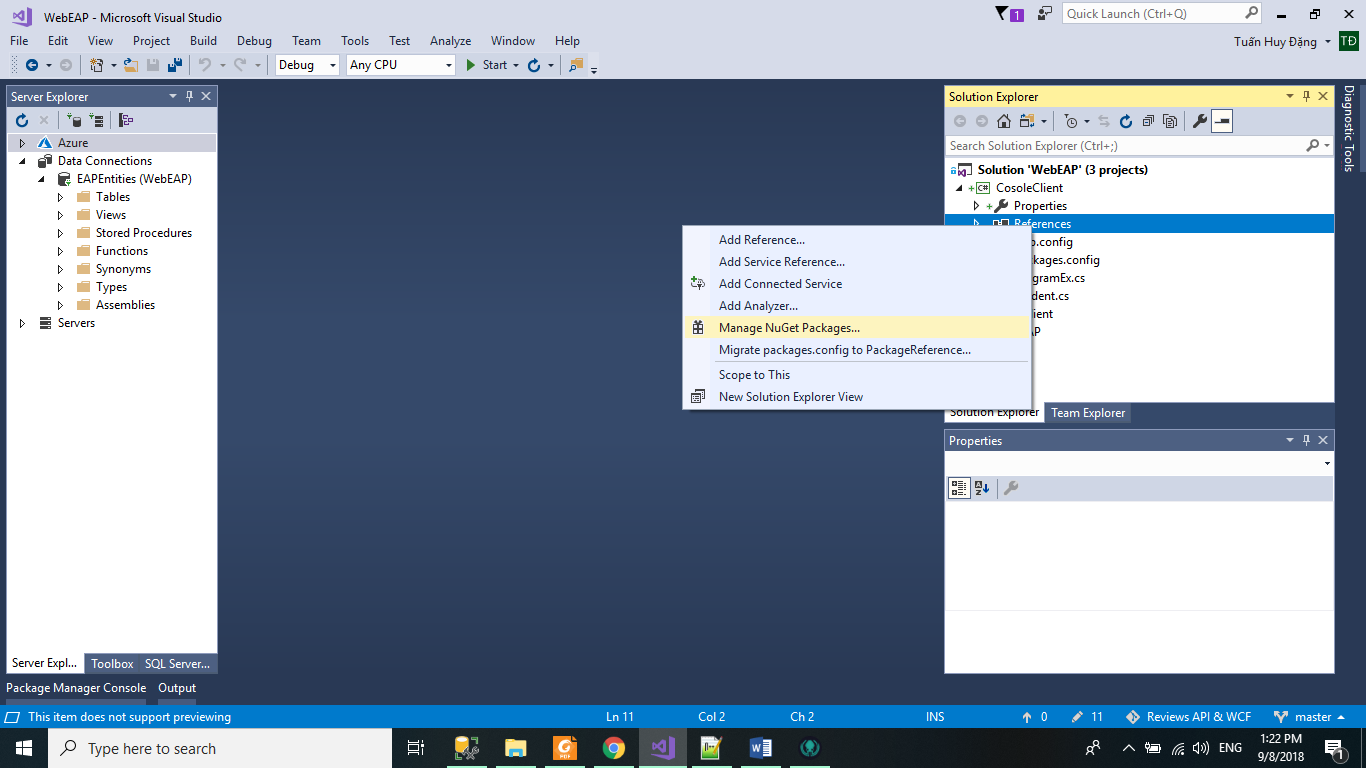


* **Edit**

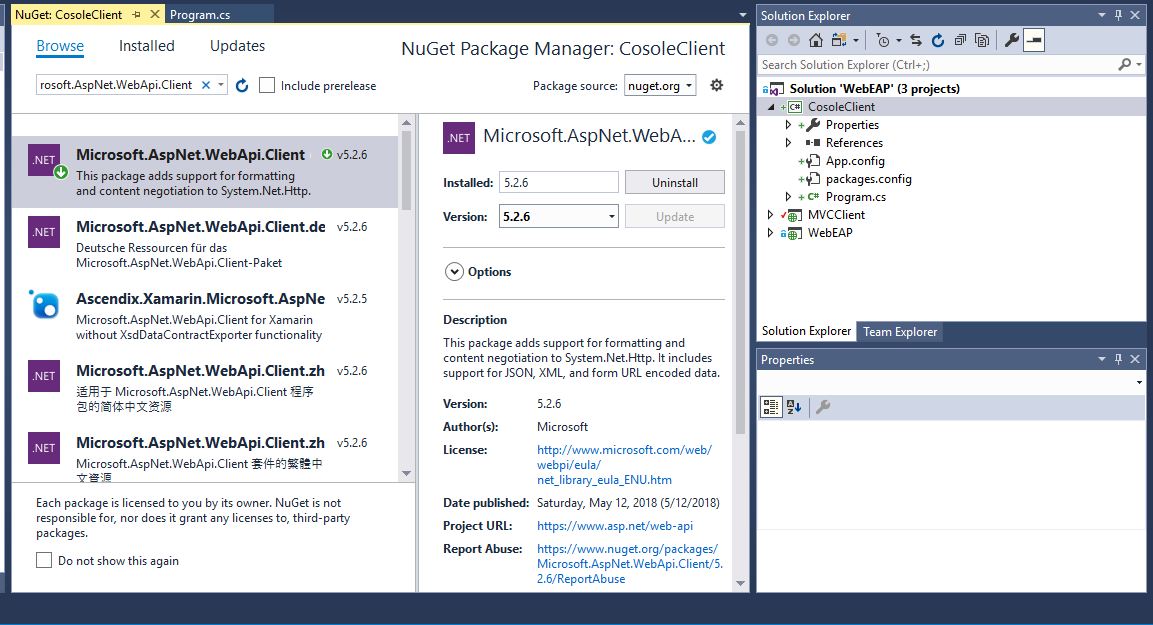


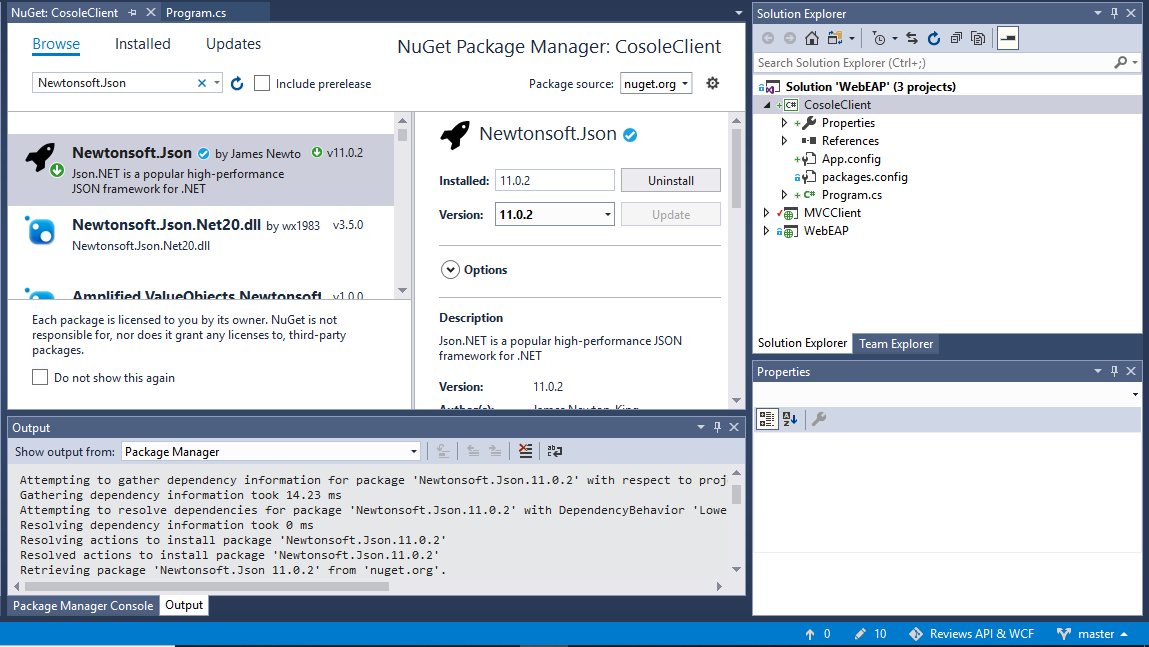
* **Delete**



1. **Console – Client**
2. **Trong cùng solution tạo một Project Console Client (Cosole Application)**
3. **Thêm hai package cho cosele App vừa tạo**

**Microsoft.AspNet.WebApi.Client**

**Newtonsoft.Json** 

1. **Tạo một Class có tên là Student.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CosoleClient

{

class Student

{

public string Student\_Code { get; set; }

public string Student\_Name { get; set; }

public string Student\_Email { get; set; }

public string Student\_Phone { get; set; }

public string Student\_Details { get; set; }

}

}

1. **Đổi tên Programs.cs thành ProgramEx.cs và bổ sung code gọi API**

using Newtonsoft.Json.Linq;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Text;

using System.Threading.Tasks;

namespace CosoleClient

{

class ProgramEx

{

// Lấy thông tin tất cả sinh viên

static async Task GetAll()

{

string url = "http://localhost:50180/api/Students";

var client = new HttpClient();

client.BaseAddress = new Uri(url);

client.DefaultRequestHeaders.Accept.Clear();

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

HttpResponseMessage response = await client.GetAsync(url);

if (response.IsSuccessStatusCode)

{

var responseData = response.Content.ReadAsStringAsync().Result;

JArray parsed = JArray.Parse(responseData.ToString());

Console.WriteLine("------------------------List of Student ' Information-------------");

foreach (var pair in parsed)

{

JObject obj = JObject.Parse(pair.ToString());

foreach (var s in obj)

{

Console.WriteLine("{0} : {1}", s.Key, s.Value.ToString().Trim());

}

Console.WriteLine();

}

}

}

// Tìm Kiếm sinh viên theo ID nè

static async Task GetStudent(string id)

{

string url = "http://localhost:50180/api/Students";

var client = new HttpClient();

client.BaseAddress = new Uri(url);

client.DefaultRequestHeaders.Accept.Clear();

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

HttpResponseMessage response = await client.GetAsync(url + "/" + id);

if (response.IsSuccessStatusCode)

{

Student student = await response.Content.ReadAsAsync<Student>();

Console.WriteLine("--------------------Lecturer information--------------------");

Console.WriteLine("Student Code: " + id);

Console.WriteLine("Student Name: " + student.Student\_Name);

Console.WriteLine("Student Email: " + student.Student\_Email);

Console.WriteLine("Student Phone: " + student.Student\_Phone);

Console.WriteLine("Student Details: " + student.Student\_Details.ToString().Trim());

}

else

{

Console.WriteLine("Lecturer is not exist");

}

}

static async Task AddNewStudent(string id, string name, string email, string phone, string details)

{

string url = "http://localhost:50180/api/Students";

var client = new HttpClient();

client.BaseAddress = new Uri(url);

client.DefaultRequestHeaders.Accept.Clear();

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

HttpResponseMessage response = await client.GetAsync(url);

Student st = new Student()

{

Student\_Code = id,

Student\_Name = name,

Student\_Email = email,

Student\_Phone = phone,

Student\_Details = details

};

response = await client.PostAsJsonAsync(url, st);

if (response.IsSuccessStatusCode)

{

Console.WriteLine("--------------------Student's information is added--------------------");

}

else

{

Console.WriteLine("--------------------Student's information cannot addto database--------------------");

}

}

static async Task UpdateStudent(string id, string name, string email, string phone, string details)

{

string url = "http://localhost:50180/api/Students";

var client = new HttpClient();

client.BaseAddress = new Uri(url);

client.DefaultRequestHeaders.Accept.Clear();

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

HttpResponseMessage response = await client.GetAsync(url + "/" + id);

if (response.IsSuccessStatusCode)

{

Student student = await response.Content.ReadAsAsync<Student>();

if (name != "")

student.Student\_Name = name;

if (email != "")

student.Student\_Email = email;

if (phone != "")

student.Student\_Phone = phone;

if (details != "")

student.Student\_Details = details;

response = await client.PutAsJsonAsync(url + "/" + id, student);

if (response.IsSuccessStatusCode)

{

Console.WriteLine("--------------------Update Student Successfully--------------------");

}

else

{

Console.WriteLine("--------------------Update Failure--------------------");

}

}

}

static async Task DeleteStudent(string id)

{

string url = "http://localhost:50180/api/Students";

var client = new HttpClient();

client.BaseAddress = new Uri(url);

client.DefaultRequestHeaders.Accept.Clear();

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

HttpResponseMessage response = await client.GetAsync(url + "/" + id);

response = await client.DeleteAsync(url + "/" + id);

if (response.IsSuccessStatusCode)

{

Console.WriteLine("--------------------Delete Successfully--------------------");

}

else

{

Console.WriteLine("--------------------Delete Failure--------------------");

}

}

static void Main(string[] args)

{

string i;

do

{

Console.WriteLine("1. Get all Student' information");

Console.WriteLine("2. Get only one student's information");

Console.WriteLine("3. Add new student");

Console.WriteLine("4. Update student's information");

Console.WriteLine("5. Delete student from database");

Console.Write("Enter your choice: ");

int ans = Convert.ToInt32(Console.ReadLine());

if (ans == 1)

{

GetAll().Wait();

}

else

{

if (ans == 2)

{

Console.Write("Enter the student code: ");

string id = Console.ReadLine();

GetStudent(id).Wait();

}

else

{

if (ans == 3)

{

Console.WriteLine("--------------------Enter the Student'sInformation--------------------");

Console.Write("Enter Student Code:");

string id = Console.ReadLine();

Console.Write("Enter Student Name:");

string name = Console.ReadLine();

Console.Write("Enter Student Email:");

string email = Console.ReadLine();

Console.Write("Enter Student Phone:");

string phone = Console.ReadLine();

Console.Write("Enter Student Details:");

string details = Console.ReadLine();

AddNewStudent(id, name, email, phone, details).Wait();

}

else

{

if (ans == 4)

{

Console.Write("Enter the student Code: ");

string id = Console.ReadLine();

Console.WriteLine("--------------------Enter the Student's Information--------------------");

Console.Write("Enter student Name:");

string name = "";

name = Console.ReadLine();

Console.Write("Enter studentEmail:");

string email = "";

email = Console.ReadLine();

Console.Write("Enter student Phone:");

string phone = "";

phone = Console.ReadLine();

Console.Write("Enter student details :");

string details = "";

details = Console.ReadLine();

UpdateStudent(id, name, email, phone, details).Wait();

}

else

{

Console.Write("Enter the student code: ");

string id = Console.ReadLine();

DeleteStudent(id).Wait();

}

}

}

}

Console.Write("Do you want to continue: ");

i = Console.ReadLine();

} while (i == "y" || i == "Y");

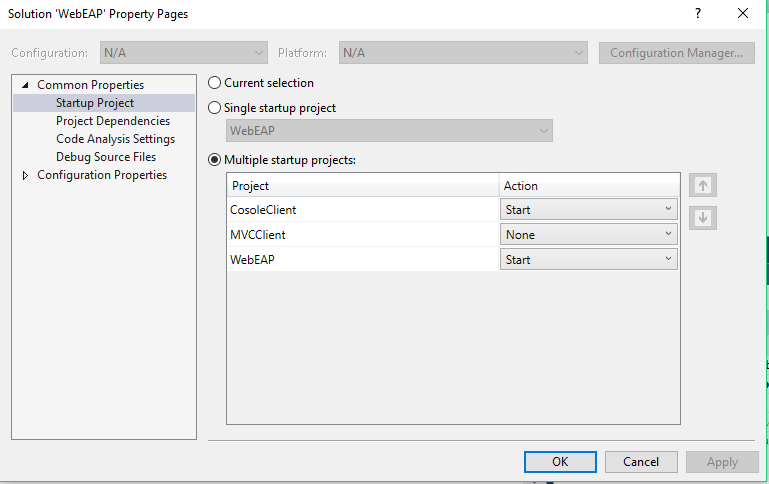
Console.ReadLine();

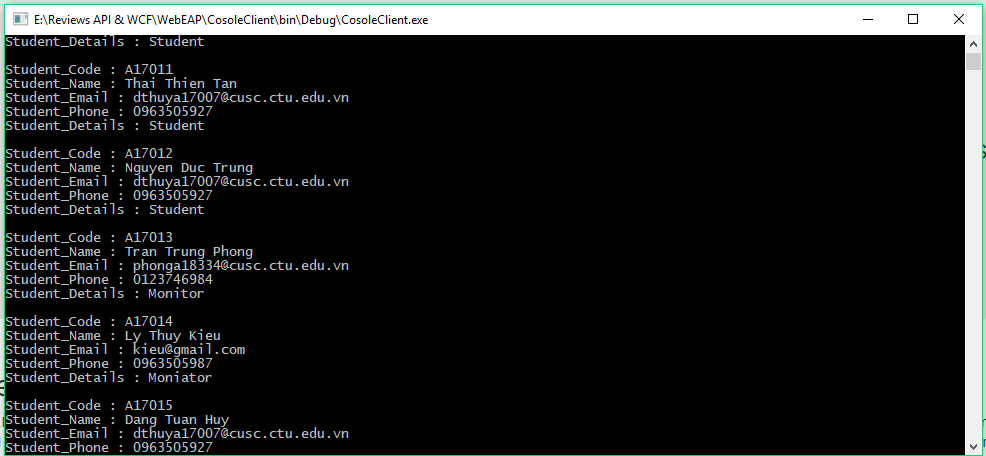
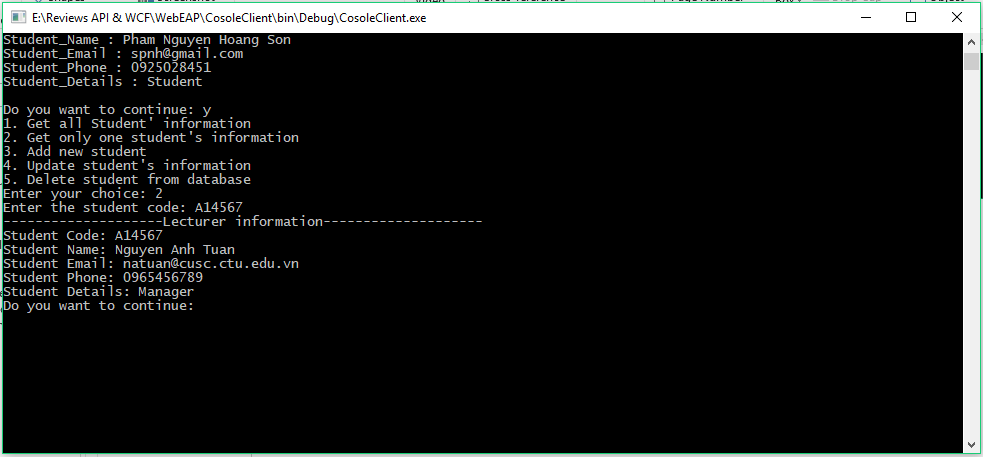
}

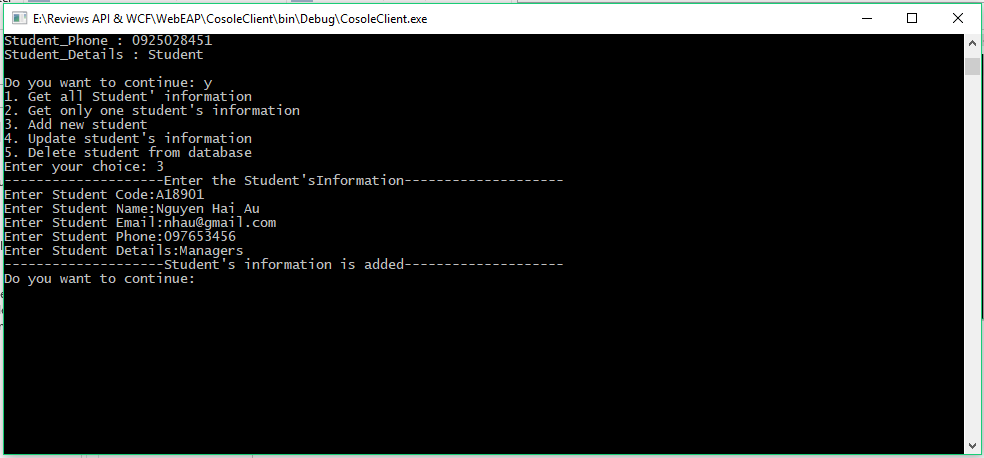
}

}

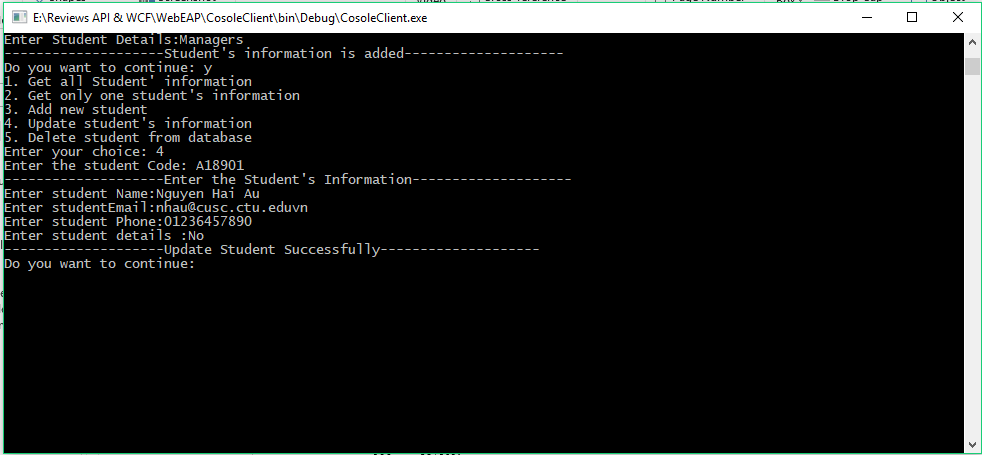
1. **Kết quả:**

**Config App:**

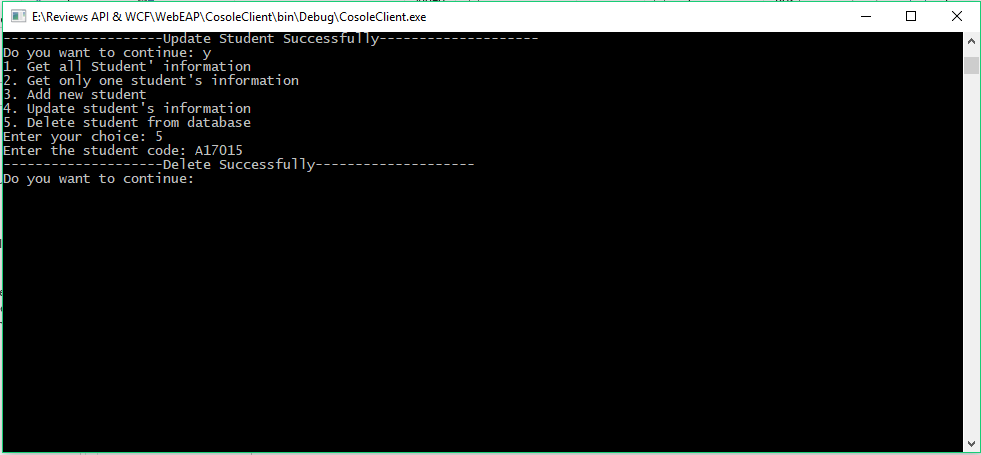
* **Danh sách toàn bộ**
* **Tìm theo ID**
* **Thêm**

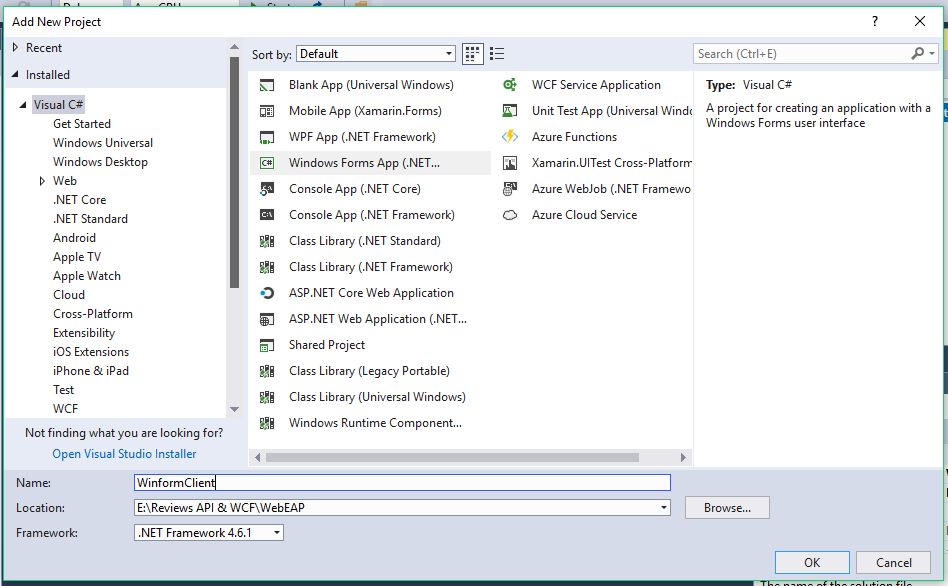
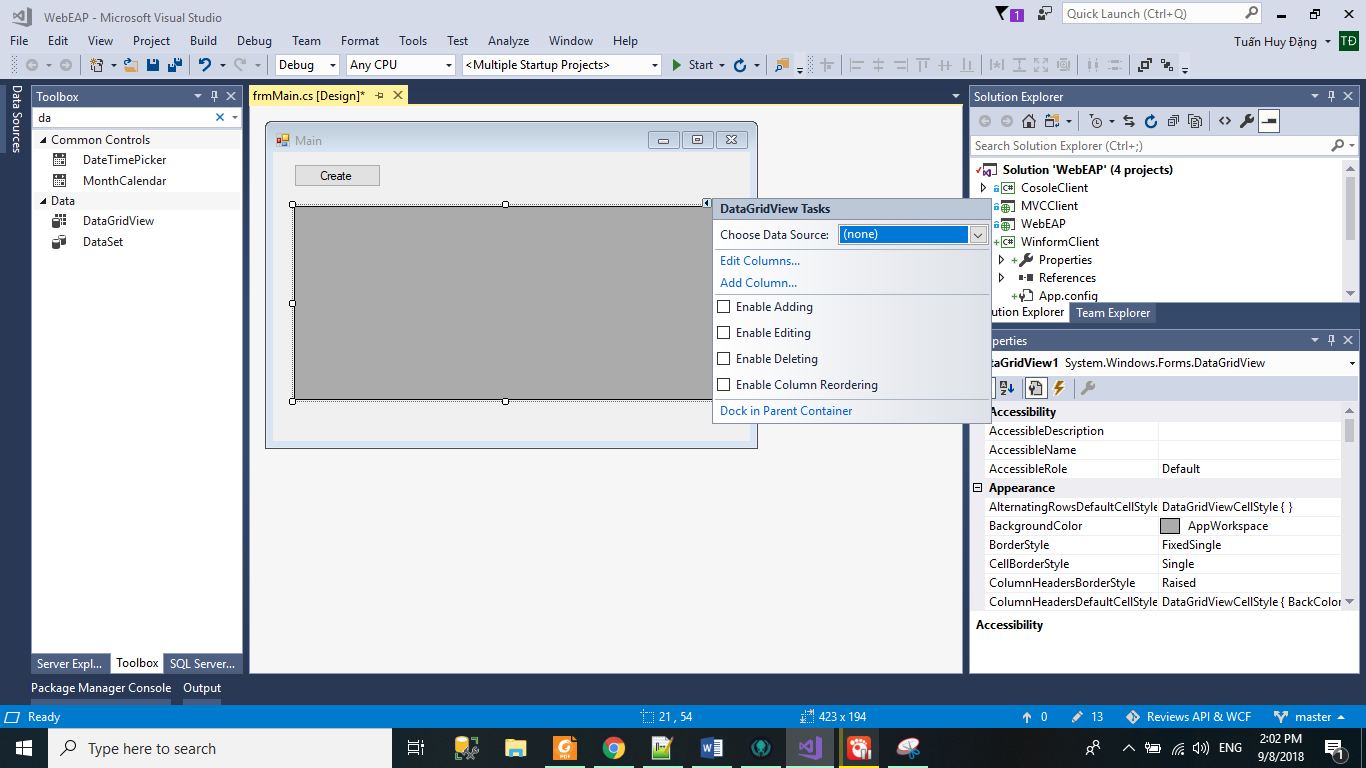


* **Sửa**

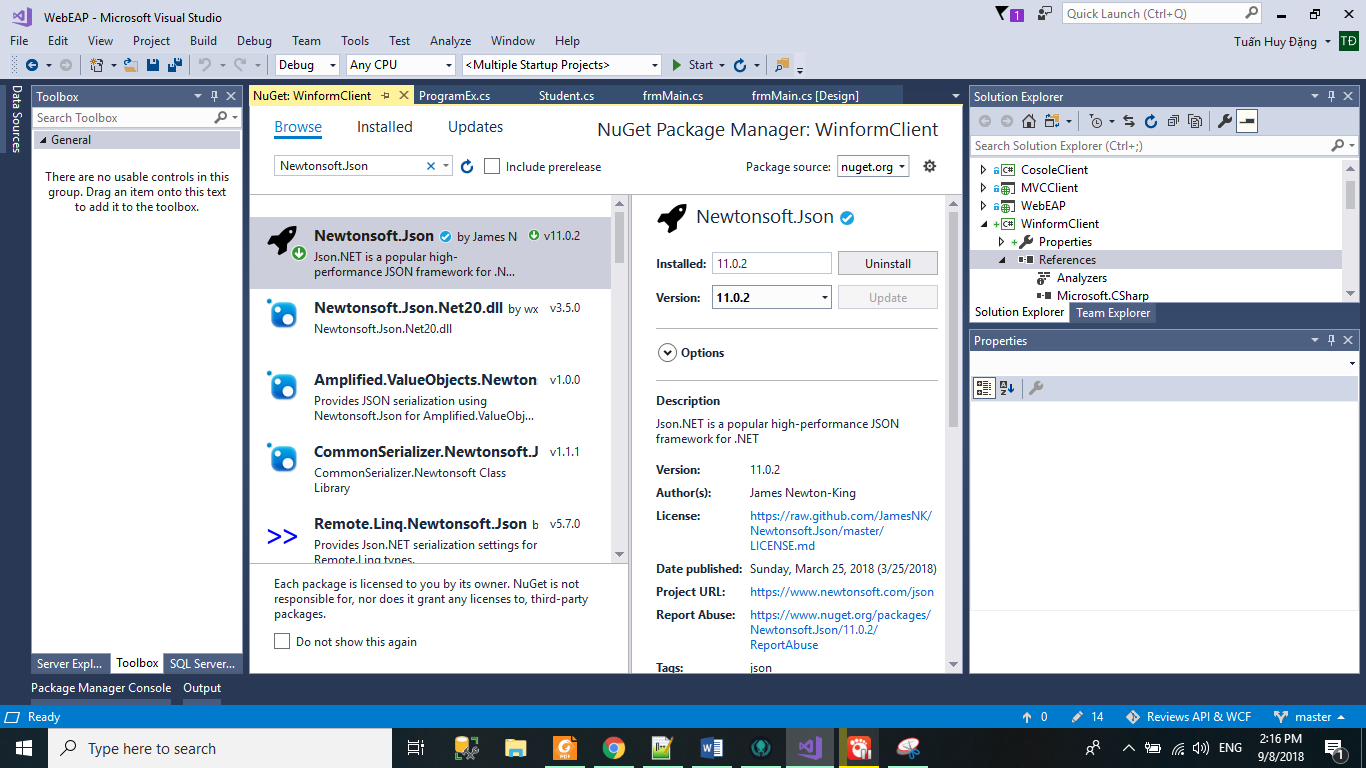
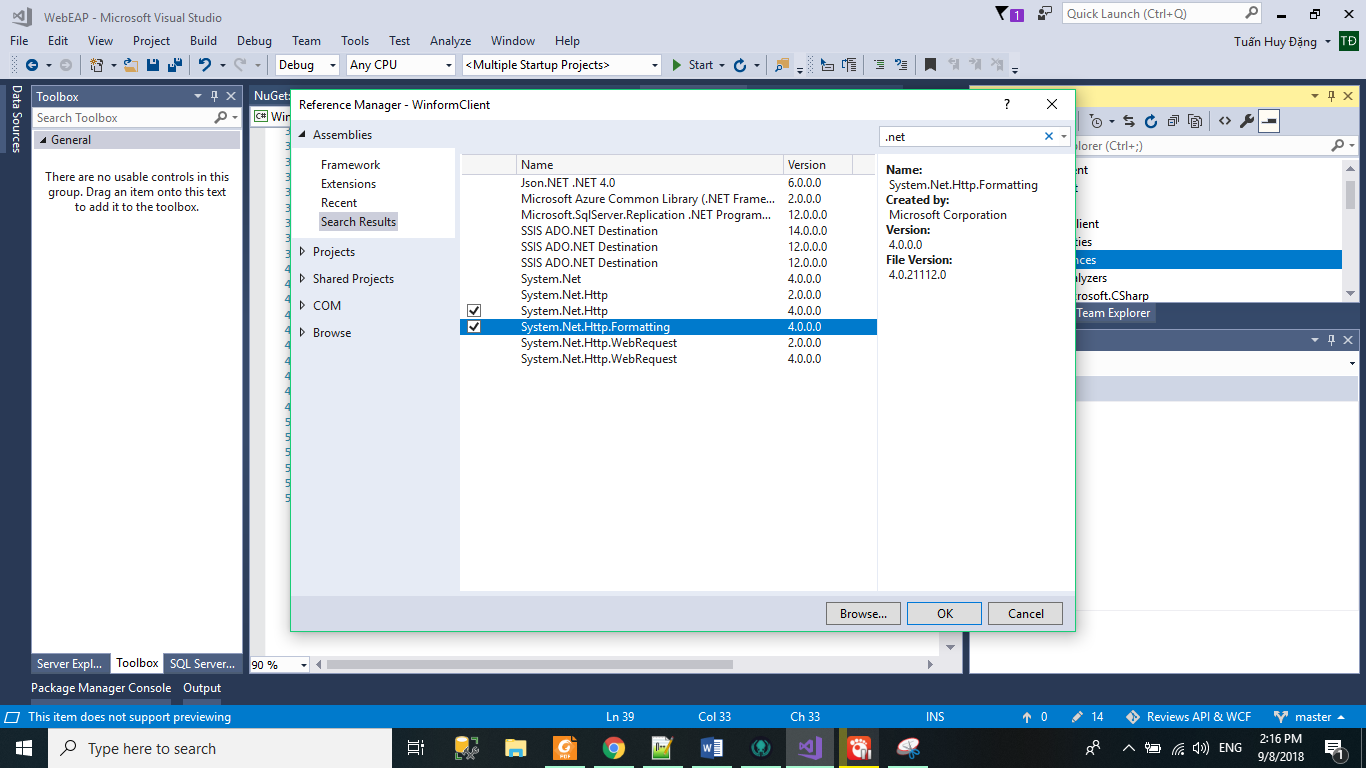


* **Xóa**



1. **WindowForm – Client**
2. **Trong cùng một solution tạo một Project Window Form Application**
3. **Tạo form sau**

**Button : Name btnCreate**

1. **Install Pakage**
2. **Config**
3. **WCF**