

สร้าง web service



Step 1



Visual Studio 2022

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Start learning

Learn the basics, explore settings, or create projects step-by-step with guides



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

[Continue without code](#) →





Step 2



Create a new project

Recent project templates

 WCF Service Application C#

 ASP.NET Web Application
(.NET Framework) C#

WCF

All languages

All platforms

All project types



WCF Service

A Web site for creating WCF services. This template does not produce a project file and has limited MSBuild support.

C#

Windows

Web

Service



WCF Service Application

A project for creating WCF Service Application that is hosted in IIS/WAS

C#

Windows

Web

Service



WCF Service Library

A project for creating a host-independent WCF service class library (.dll)

C#

Windows

Library

Web

Service



WCF Service Application

A project for creating WCF Service Application that is hosted in IIS/WAS

Visual Basic

Windows

Web



WCF Service Library

A project for creating a host-independent WCF service class library (.dll)

Back

Next

Step 3



Configure your new project

WCF Service Application

C#

Windows

Web

Service

Project name

WcfService1

Location

C:\Users\khopo\source\repos

Solution name ⓘ

WcfService1

☐ Place solution and project in the same directory

Framework

.NET Framework 4.7.2

Back

Create

Step 4



Visual Studio interface showing a WCF service project. The code editor displays the following code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Runtime.Serialization;
5 using System.ServiceModel;
6 using System.ServiceModel.Web;
7 using System.Text;
8
9 namespace WcfService1
10 {
11     // NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name.
12     // NOTE: In order to launch WCF Test Client for testing this service, please select the service from the list.
13     public class Service1 : IService1
14     {
15         public string GetData(int value)
16         {
17             return string.Format("You entered: {0}", value);
18         }
19
20         public CompositeType GetDataUsingDataContract(CompositeType composite)
21         {
22             if (composite == null)
23             {
24                 // ...
25             }
26         }
27     }
28 }
```

The right sidebar shows the 'Add' menu, which is open. The 'Add' option is highlighted. The 'Add' menu includes the following options:

- New Item... (Ctrl+Shift+A)
- Existing Item... (Shift+Alt+A)
- New Scaffolded Item...
- New Folder
- Add ASP.NET Folder
- Container Orchestrator Support...
- Docker Support...
- Application Insights Telemetry...
- REST API Client...
- Client-Side Library...
- New Azure WebJob Project
- Existing Project as Azure WebJob
- Reference...
- Service Reference...
- Connected Service
- Analyzer...
- HTML Page
- JavaScript File
- Style Sheet
- Web Form
- MVC 5 View Page (Razor)
- Web API Controller Class (v2.1)
- Web Service (ASMX)
- Class...
- New EditorConfig

The bottom status bar shows 'Ready'.

Step 5



Add New Item - WcfService1

Installed

Sort by: Default

C#

- Code
- Data
- General
- Web
- General
- Markup
- Scripts
- Web Forms
- MVC
- Razor
- SignalR
- Web API
- SQL Server

Web Service (ASMX)

Type: C#

A visually designed class for creating a Web Service

Name: MathService.asmx

Add Cancel

Step 6



Visual Studio interface showing the development of a WCF service. The main editor displays the code for `MathService.asmx.cs`, which is part of the `WcfService1` project. The code defines a `WebService1` class that inherits from `System.Web.Services.WebService`. It includes two web methods: `Add` and `Subtract`.

```
using System.Web.Services;

namespace WcfService1
{
    /// <summary>
    /// Summary description for WebService1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
    // [System.Web.Script.Services.ScriptService]
    public class WebService1 : System.Web.Services.WebService
    {
        [WebMethod]
        public double Add(double a, double b)
        {
            return (a + b);
        }

        [WebMethod]
        public double Subtract(double a, double b)
        {
            return (a - b);
        }
    }
}
```

The Solution Explorer on the right shows the project structure, with `MathService.asmx` highlighted. The Properties window is empty. The Output window at the bottom shows "No issues found".

```
[WebMethod]
public double Add(double a,
double b)
{
    return (a + b);
}
```

```
[WebMethod]
public double
Subtract(double a, double b)
{
    return (a - b);
}
```



```
[WebMethod]
public double
Multiply(double a, double b)
{
    return a * b;
}
```

```
[WebMethod]
public double Divide(double
a, double b)
{
    if (b == 0)
        return -1;
    return a / b;
}
```


สร้าง web page เพื่อทดสอบ web service





Step 1



Create a new project

Recent project templates

-  WCF Service Application C#
-  ASP.NET Web Application (.NET Framework) C#

ASP.NET web application

Clear all

All languages

All platforms

All project types



ASP.NET Web Application (.NET Framework)

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

C#

Windows

Cloud

Web



ASP.NET Web Application (.NET Framework)

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Visual Basic

Windows

Cloud

Web



ASP.NET Core Web App

A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content

C#

Linux

macOS

Windows

Cloud

Service

Web



ASP.NET Core Web API

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

C#

Linux

macOS

Windows

Cloud

Service

Web

WebAPI

Back

Next

Step 2



Configure your new project

ASP.NET Web Application (.NET Framework)

C#

Windows

Cloud

Web

Project name

TestWebService

Location

C:\Users\khopo\source\repos

Solution name ⓘ

TestWebService

☐ Place solution and project in the same directory

Framework

.NET Framework 4.7.2

Back

Create

Step 3



Create a new ASP.NET Web Application



Empty

An empty project template for creating ASP.NET applications. This template does not have any content in it.



Web Forms

A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.



MVC

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.



Web API

A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.



Single Page Application

A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and

Authentication

None

Add folders & core references

- ☐ Web Forms
- ☐ MVC
- ☐ Web API

Advanced

- ☒ Configure for HTTPS
- ☐ Docker support
(Requires [Docker Desktop](#))
- ☐ Also create a project for unit tests

TestWebService.Tests

Back

Create

Step 4



File

Edit

View

Git

Project

Build

Debug

Test

Analyze

Tools

Extensions

Window

Help

Search (Ctrl+Q)

Debug

Any CPU

IIS Express (Google Chrome)

TestWebService: Overview

Overview

Connected Services

Publish

ASP.NET

Learn about the .NET platform, create your first application and

Build Your App

Get started with ASP.NET

.NET application architecture

Connect To The Cloud

Publish your app to Azure

Get started with ASP.NET on Azure

Output

Show output from:

Creating project 'TestWebService'... project creation successful.

New Item...

Ctrl+Shift+A

Existing Item...

Shift+Alt+A

New Scaffolded Item...

New Folder

Add ASP.NET Folder

Container Orchestrator Support...

Docker Support...

Application Insights Telemetry...

REST API Client...

Client-Side Library...

New Azure WebJob Project

Existing Project as Azure WebJob

Reference...

Service Reference...

Connected Service

Analyzer...

HTML Page

JavaScript File

Style Sheet

Web Form

MVC 5 View Page (Razor)

Web API Controller Class (v2.1)

Web Service (ASMX)

Class...

New EditorConfig

Build

Rebuild

Clean

View

Analyze and Code Cleanup

Publish...

Configure Application Insights...

Overview

Scope to This

New Solution Explorer View

Add

Manage NuGet Packages...

Manage Client-Side Libraries...

Manage User Secrets

Set as Startup Project

Debug

Initialize Interactive with Project

Cut

Ctrl+X

Paste

Ctrl+V

Remove

Del

Rename

F2

Unload Project

Load Direct Dependencies

Load Entire Dependency Tree

Copy Full Path

Open Folder in File Explorer

Open in Terminal

Properties

Alt+Enter

Step 5



Add New Item - TestWebService

Installed

Sort by: Default

C#

- Code
- Data
- General
- Web
- General
- Markup
- Scripts
- Web Forms
- MVC
- Razor
- SignalR
- Web API
- SQL Server

Online

Icon	Item Name	Type
	HTML Page	C#
	JavaScript File	C#
	Style Sheet	C#
	Web Form	C#
	Web Form with Master Page	C#
	MVC 5 View Page (Razor)	C#
	MVC 5 View Page with Layout (Razor)	C#
	Web API Controller Class (v2.1)	C#
	SignalR Hub Class (v2)	C#
	SignalR Persistent Connection Class (v2)	C#

Type: C#
A form for Web Applications

Name: TestMathService.aspx

Add Cancel



The screenshot shows the Visual Studio 2019 interface. The main editor displays the source code of `TestMathService.aspx`, which is an ASP.NET Web Form. The code includes a page directive for C#, a title, and a form with a single text input field. The `Page` attribute is configured with `Language="C#" AutoEventWireup="true" CodeBehind="TestMathService.aspx.cs" Inherits="TestMathService"`. The Form has an ID of form1 and is set to runat="server". The Page attribute also includes DefaultPageFile="TestMathService.aspx".`

The `TestWebService: Overview` tab is active, showing a summary of the project. The `Output` window at the bottom shows the message: "This item does not support previewing".

The `Add` menu is open, showing various options for adding new items to the project. The `Service Reference...` option is highlighted, indicating the next step in the process of adding a service reference.

Step 7



WebService1 Web Service

localhost:58427/MathService.as

Facebook Twitter YouTube Google แปลภาษา Google

WebService1

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [Add](#)
- [Divide](#)
- [Multiply](#)
- [Subtract](#)

Step 8



Add Service Reference

?

×

To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

Address:

http://localhost:58427/MathService.asmx

Go

Discover

Services:

Operations:

WebService1

Select a service contract to view its operations.

1 service(s) found at address 'http://localhost:58427/MathService.asmx'.

Namespace:

ServiceReference1

Advanced...

OK

Cancel

Step 9



Service Reference Settings

Client _____

Access level for generated classes: Public

☒ Allow generation of asynchronous operations

☒ Generate task-based operations

☐ Generate asynchronous operations

Data Type _____

☐ Always generate message contracts

Collection type: System.Array

Dictionary collection type: System.Collections.Generic.Dictionary

☒ Reuse types in referenced assemblies

☒ Reuse types in all referenced assemblies

☐ Reuse types in specified referenced assemblies:

- ☐ Microsoft.CodeDom.Providers.DotNetCompilerPlatform
- ☐ Microsoft.CSharp
- ☐ mscorlib
- ☐ System
- ☐ System.ComponentModel.DataAnnotations
- ☐ System.Configuration

Compatibility _____

Add a Web Reference instead of a Service Reference. This will generate code based on .NET Framework 2.0 Web Services technology.

Add Web Reference...

OK Cancel

Step 10



Add Web Reference

Navigate to a web service URL and click Add Reference to add all the available services.



URL: →

WebService1

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [Add](#)
- [Divide](#)
- [Multiply](#)
- [Subtract](#)

This web service is using `http://tempuri.org/` as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. `http://tempuri.org/` is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

Web services found at this URL:

1 Service Found:

- MathService

Web reference name:

Add Reference

Cancel

Step 11



Visual Studio IDE interface showing the 'TestMathService.aspx' file in the Design view. The 'Design' tab is selected and highlighted with a red box. The 'Output' window at the bottom shows debug messages from 'iisexpress.exe' and a message indicating the program has exited with code 4294967295 (0xffffffff). The 'Solution Explorer' on the right shows the project structure for 'TestWebService', including 'Connected Services', 'Properties', 'AssemblyInfo.cs', 'Settings.settings', 'References', 'Web References', 'localhost', and 'packages.config'. The 'Properties' window on the right shows the 'DOCUMENT' properties for 'ASP.NET', including 'Culture', 'Debug', 'EnableSessionState', 'Language' (set to 'C#'), 'MasterPageFile', and 'StyleSheetTheme'.

Ready

Add to Source Control Select Repository

Step 12



Visual Studio interface showing the design of a web application. The main window displays the design view of `TestMathService.aspx`. The design includes two text boxes labeled `txtFirst` and `txtSecond`, a label `lblResult` with the text "= Result here!", and four buttons labeled `btnAdd`, `btnSub`, `btnMul`, and `btnDiv`. The buttons are arranged in a row below the text boxes. The `btnAdd` button is highlighted with a blue selection box.

The **Toolbox** is open on the right side of the screen, showing a list of web parts and AJAX extensions. The **WebParts** section is expanded, and the **AppearanceEditorPart** is selected. The **AJAX Extensions** section is also expanded, showing a list of controls including `ScriptManager`, `ScriptManagerProxy`, `Timer`, `UpdatePanel`, and `UpdateProgress`.

The **Output** window at the bottom shows the following message:

```
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-133151376095824543): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\...'  
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-133151376095824543): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\...'  
The program '[23624] iisexpress.exe' has exited with code 4294967295 (0xffffffff).
```

Step 13



Visual Studio interface showing the development of a web application. The main window displays the **TestMathService.aspx** file in Design view. A context menu is open over a **Button** control, with the **Properties** option highlighted. The menu includes options like Cut, Copy, Paste, Delete, View Code, View in Browser (Google Chrome), Show Smart Tag, Refresh, and Properties (Alt+Enter).

The bottom status bar shows the **Output** window with the following text:

```
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-133151376095824543): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.ServiceModel.Internals\v4.0.0.0__31bf3856ad36443f\System.ServiceModel.Internals.dll'.
'iisexpress.exe' (CLR v4.0.30319: /LM/W3SVC/2/ROOT-1-133151376095824543): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\SMDiagnostics\v4.0.0.0__b77a5c561934e089\SMDiagnostics.dll'.
The program '[23624] iisexpress.exe' has exited with code 4294967295 (0xffffffff).
```

The bottom status bar also shows "Ready" and options to "Add to Source Control" and "Select Repository".

Step 14



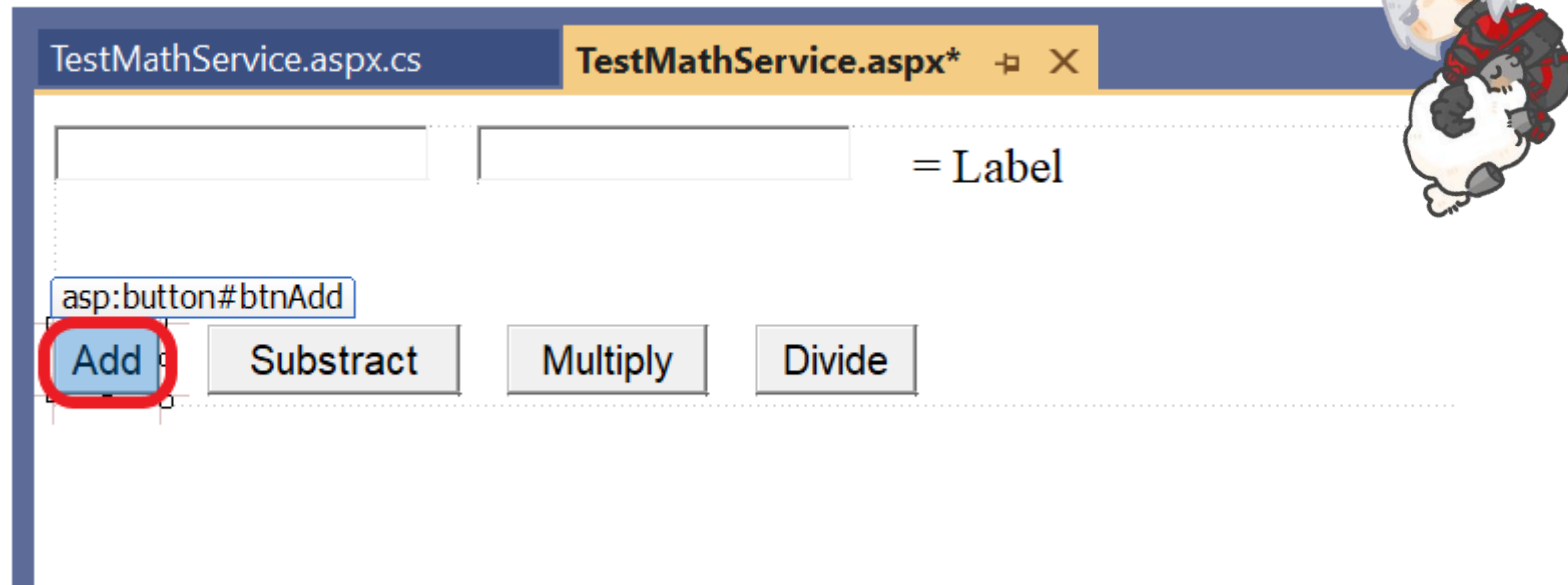
Visual Studio interface showing the design of a web application. The main window displays the design view of `TestMathService.aspx`. The design view shows a form with two input fields and a label `= Label`. Below the input fields, there are four buttons: `asp:button#btnAdd` (highlighted with a red circle), `Button`, `Button`, and `Button`.

The Properties window on the right shows the properties for the selected control, `btnAdd` (System.Web.UI.WebControls.Button). The properties are categorized into Accessibility, Appearance, Font, Behavior, Data, Layout, and Misc. The `Text` property is highlighted with a red circle and the value `Add`. The `(ID)` property is highlighted with a red circle and the value `btnAdd`.

The Output window at the bottom shows the debug output from `iisexpress.exe`. The output indicates that the program has exited with code 4294967295 (0xffffffff).

Ready

Step 15



คลิก 2 ครั้ง เพื่อไปหน้าโค้ด

Step 16



Visual Studio interface showing the code for `TestMathService.aspx.cs`. The code is highlighted with a red box.

```
10 public partial class TestMathService : System.Web.UI.Page
11 {
12     protected void Page_Load(object sender, EventArgs e)
13     {
14     }
15 }
16
17 protected void btnAdd_Click(object sender, EventArgs e)
18 {
19     localhost.MathService myMathService = new localhost.MathService();
20     double a = Convert.ToDouble(txtFirst.Text);
21     double b = Convert.ToDouble(txtSecond.Text);
22     double result = myMathService.Add(a, b);
23     lblResult.Text = Convert.ToString(result);
24 }
25
26
```

The Solution Explorer on the right shows the project structure for `TestWebService`, including `AssemblyInfo.cs`, `Settings.settings`, `References`, and `Web References` (localhost).

The Output window at the bottom shows "No issues found".

Item(s) Saved

Add to Source Control Select Repository

```
protected void btnAdd_Click1(object sender, EventArgs e)
{
    localhost.MathService myMathService = new
localhost.MathService();
    double a = Convert.ToDouble(txtFirst.Text);
    double b = Convert.ToDouble(txtSecond.
Text);
    double result = myMathService.Add(a,b);
    lblResult.Text = Convert.ToString(result);
}

protected void btnSub_Click(object sender, EventArgs e)
{
    localhost.MathService myMathService = new
localhost.MathService();
    double a = Convert.ToDouble(txtFirst.Text);
    double b = Convert.ToDouble(txtSecond.Text);
    double result = myMathService.Subtract(a, b);
    lblResult.Text = Convert.ToString(result);
}
```



```
protected void btnMul_Click(object sender, EventArgs e)
{
    localhost.MathService myMathService = new
localhost.MathService();
    double a = Convert.ToDouble(txtFirst.Text);
    double b = Convert.ToDouble(txtSecond.Text);
    double result = myMathService.Multiply(a, b);
    lblResult.Text = Convert.ToString(result);
}

protected void btnDiv_Click(object sender, EventArgs e)
{
    localhost.MathService myMathService = new
localhost.MathService();
    double a = Convert.ToDouble(txtFirst.Text);
    double b = Convert.ToDouble(txtSecond.Text);
    double result = myMathService.Divide(a, b);
    lblResult.Text = Convert.ToString(result);
}
```

Show

24

24

= 48

Add

Subtract

Multiply

Divide



Ex. สร้าง web service แปลงอุณหภูมิแปลง

1. Celsius เป็น Fahrenheit
2. Celsius เป็น Kelvin
3. Fahrenheit เป็น Celsius
4. Fahrenheit เป็น Kelvin
5. Kelvin เป็น Celsius
6. Kelvin เป็น Fahrenheit

Web Service : Temperature

Web Page : TestTemp

[WebMethod] (ใส่ทุกครั้งห้ามลืม)



Step 1

```
[WebMethod]
public double CtoK(double c)
{
    double k = c + 273.15;
    return k;
}
```

```
[WebMethod]
public double KtoC(double k)
{
    double c = k - 273.15;
    return c;
}
```

```
[WebMethod]
public double FtoK(double f)
{
    double k = ((f - 32) * 5 / 9) + 273.15;
    return k;
}
```

```
[WebMethod]
public double KtoF(double k)
{
    double f = ((k - 273.15) * 9 / 5) + 32;
    return f;
}
```



Step 1

[WebMethod]

public double CtoF(double c)

{

double f = (c * 9 / 5) + 32;

return f;

}

[WebMethod]

public double FtoC(double f)

{


double c = (f - 32) * 5 / 9;

return c;

}



Step 2 : Web Page (Design)



TestTemp.aspx

☐ CtoK ☐ KtoC ☐ FtoK ☐ KtoF ☐ CtoF ☐ FtoC

Convert

Result Here!



ID

Radio Button ชื่อเดียวกับ text

Text Box ชื่อ txt1

Label ชื่อ lblresult

Step 3 : เอา code ใส่ button

```
protected void btnConvert_Click(object sender, EventArgs e)
{
    if (CtoK.Checked == true)
    {
        lblresult.Text = Convert.ToString((Convert.ToInt16(txt1.Text) + 273.15));
    }
    else if (KtoC.Checked == true)
    {
        lblresult.Text = Convert.ToString((Convert.ToInt16(txt1.Text) - 273.15));
    }
}
```



Step 3 : เอา code ใส่ button

```
else if (FtoK.Checked == true)
{
    lblresult.Text = Convert.ToString(((Convert.ToInt16(txt1.Text) - 32) * 5 / 9) + 273.15);
}
else if (KtoF.Checked == true)
{
    lblresult.Text = Convert.ToString(((Convert.ToInt16(txt1.Text) - 273.15) * 9 / 5) + 32);
}
```



Step 3 : เอา code ใส่ button

```
else if (CtoF.Checked == true)
{
    lblresult.Text = Convert.ToString((Convert.ToInt16(txt1.Text) * 9 / 5) + 32);
}
else if (FtoC.Checked == true)
{
    lblresult.Text = Convert.ToString((Convert.ToInt16(txt1.Text) - 32) * 5 / 9);
}
}
```



Show

☐ CtoK

☐ KtoC

☐ FtoK

☐ KtoF

☐ CtoF

☐ FtoC

Convert

Result Here!



Ex. สร้าง web service คำนวณ Quadratic

$$4x^2 + 6x + 2 = 0 \text{ --- easy}$$

$$4x^2 + 8x + 2 = 0 \text{ --- ???}$$

$$4x^2 + 6x + 2 = 0 \text{ --- easy}$$

$$(4x+2)(x+1) = 0$$

$$(4x+2)=0$$

$$x_1 = -0.5$$

$$(x+1) = 0$$

$$x_2 = -1$$

สมการ Quadratic ($ax^2 + bx + c = 0$)

$$ax^2 + bx + c = 0$$

$$x_1, x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Web Service : Quadratic

Web Page : TestQuadratic

[WebMethod] (ใส่ทุกครั้งที่ขี้ลืม)



Step 1

[WebMethod]

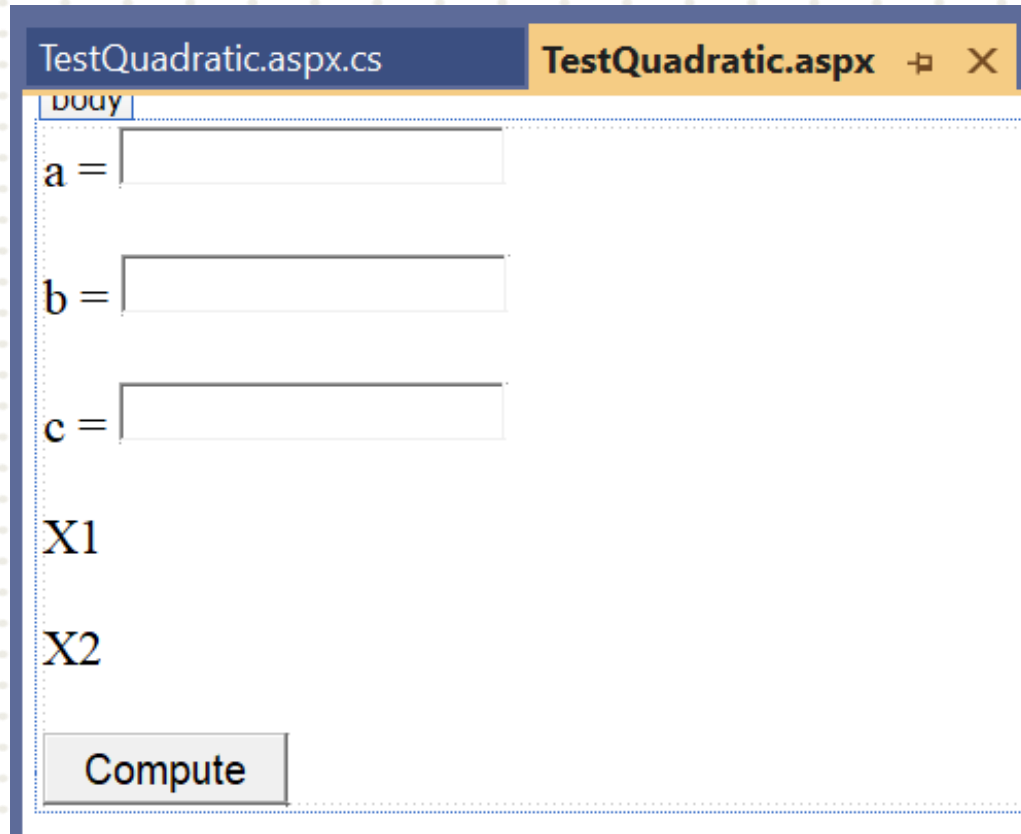
```
public double X1(double a, double b, double c)
{
    double x1 = (-b + Math.Sqrt((b*b) - (4 * a * c))) / (2 * a);
    return x1;
}
```

[WebMethod]

```
public double X2(double a, double b, double c)
{
    double x2 = (-b - Math.Sqrt((b * b) - (4 * a * c))) / (2 * a);
    return x2;
}
```



Step 2 : Web Page (Design)



The screenshot shows a web browser window with the title 'TestQuadratic.aspx'. The page content includes a label 'Body' at the top. Below it are three input fields for coefficients: 'a =', 'b =', and 'c ='. Under these fields are two labels, 'X1' and 'X2', representing the roots. At the bottom of the form is a button labeled 'Compute'.

ID

Text Box ชื่อ txtA, txtB, txtC

Label ชื่อ lblX1, lblX2

Button ชื่อ btnCompute



Step 3 : ទៅ code ទៅ button

```
protected void btnCompute_Click(object sender, EventArgs e)
{
    localhost.Quadratic myQuad = new localhost.Quadratic();

    double a = Convert.ToDouble(txtA.Text);
    double b = Convert.ToDouble(txtB.Text);
    double c = Convert.ToDouble(txtC.Text);

    double x1 = myQuad.X1(a, b, c);
    double x2 = myQuad.X2(a, b, c);

    lblX1.Text = Convert.ToString(x1);
    lblX2.Text = Convert.ToString(x2);
}
```



Show

a =

b =

c =

X1

X2

Compute



Ex. สร้าง web page : public web services

Step 1 : ไปที่เว็บไซต์ <https://www.rd.go.th/42546.html>

Step 2 : คลิก CheckTINPINService

Step 3 : จากคำว่า URL of JSON และ copy link

URL of JSON : <https://rdws.rd.go.th/jsonRD/checktinpinservice.asmx>

Step 4 : เช็ค link ที่ copy ว่าเข้าได้ไหม

Step 5 : เอา link ไปวางที่ add web reference

Web Page : TINPINService
[WebMethod] (ใส่ทุกครั้งห้ามลืม)



Step 5

Add Web Reference

?

×

Navigate to a web service URL and click Add Reference to add all the available services.

⬅️ ➡️ ■ ↺ 🏠

URL: ➡️

checktinpinservice

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [ServiceTIN](#)
Check PIN
- [ServiceTINArr](#)
Check TIN Array

Web services found at this URL:

1 Service Found:
- checktinpinservice

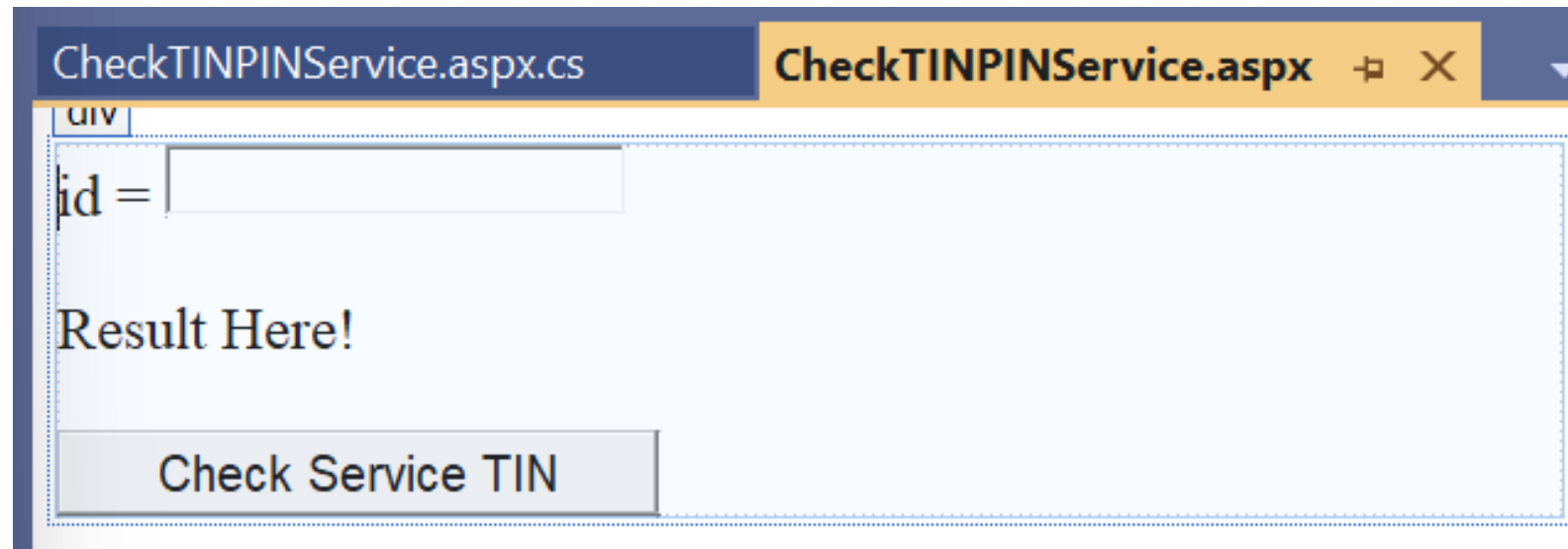
Web reference name:

Add Reference

Cancel



Step 7 : Web Page (Design)



CheckTINPINService.aspx.cs CheckTINPINService.aspx

id =

Result Here!

Check Service TIN

ID

Text Box ชื่อ txtCheckServiceTIN1

Label ชื่อ lblResult

Button ชื่อ btnCheckServiceTIN1



Step 8 : ទៅ code វ៉ៃ button

```
protected void btnCheckServiceTIN1_Click(object sender,
EventArgs e)
{
    th.go.rd.rdws.checktinpinservice myService = new
th.go.rd.rdws.checktinpinservice();
    String id = txtCheckServiceTIN1.Text;
    String temp =
Convert.ToString(myService.ServiceTIN("anonymous", "anonymous",
id));
    lblResult.Text = temp;
}
```



Show

id =

Result Here!

Check Service TIN

