

Build School 課程 – C# 基礎課程 05

Bill Chung
Build School 講師
V2021.2 新竹尖兵班

請尊重講師的著作權及智慧財產權!

Build School 課程之教材、程式碼等、僅供課程中學習用、請不要任意自行散佈、重製、分享，謝謝

再論 Method

方法多載 (overloading)

- 同樣的方法名稱，不同的參數清單
 - 型別不同 或 參數數量不同
- 不可以單純只多載回傳值
- 又稱為【同名異式】
- 還記得第一天的【論 + 運算子】嗎？

合法的多載

```
static int Add(int x, int y)
{ return x + y; }
```

```
static int Add(int x, int y, int z)
{ return x + y + z; }
```

```
static double Add(double x, double y)
{ return x + y; }
```

```
static string Add(string x, string y)
{ return x + y; }
```

```
static string Add(string x, int y)
{ return x + "整數" + y.ToString(); }
```

不合法的多載

```
static void Add()  
{  
}
```

```
static int Add()  
{  
}
```

```
static void Sub(int x)  
{  
}
```

```
static int Sub(int y)  
{  
}
```

LAB

方法多載

新增一個方案及專案

- 方案名稱： MethodOverloadingSamples
- 專案名稱：MethodOverloadingSample001
- 範本：Console Application

在 Program Class 加入以下的多載方法

```
static int Add(int x, int y)
{ return x + y; }
```

```
static int Add(int x, int y, int z)
{ return x + y + z; }
```

```
static double Add(double x, double y)
{ return x + y; }
```

```
static string Add(string x, string y)
{ return x + y; }
```

```
static string Add(string x, int y)
{ return x + "整數" + y.ToString(); }
```

直接在 Main Method 加入 程式碼

```
static void Main(string[] args)
{
    Console.WriteLine(Add(1, 1));
    Console.WriteLine(Add(1, 2, 3));
    Console.WriteLine(Add(1.5, 3.2));
    Console.WriteLine(Add(9.8, 7));
    Console.WriteLine(Add("A", "B"));
    Console.WriteLine(Add("XYZ", 100));
    Console.ReadLine();
}
```

請將游標移到每個 Add 的呼叫上

```
static void Main(string[] args)
{
    Console.WriteLine(Add(1, 1));
    Console.WriteLine(Add(1, 2, 3));
    Console.WriteLine(Add(1.5, 3.2));
    Console.WriteLine(Add(9.8, 7));
    Console.WriteLine(Add("A", "B"));
    Console.WriteLine(Add("XYZ", 100));

    Console.ReadLine();
}
```

int Program.Add(int x, int y) (+ 4 多載)

0 個多載

```
static void Main(string[] args)
{
    Console.WriteLine(Add(1, 1));
    Console.WriteLine(Add(1, 2, 3));
    Console.WriteLine(Add(1.5, 3.2));
    Console.WriteLine(Add(9.8, 7));
    Console.WriteLine(Add("A", "B"));
    Console.WriteLine(Add("XYZ", 100));

    Console.ReadLine();
}
```

string Program.Add(string x, int y) (+ 4 多載)

在 Main Method 中隨便輸入個 Add()

```
Console.WriteLine(Add("XYZ", 100));  
Add()  
    ▲ 1 / 5 ▼ double Program.Add(double x, double y)  
Console.ReadLine();  
}
```

操作一下這兩個三角形，你看到了甚麼？
執行前請將這個 Add() 刪除。

執行

討論

- 甚麼條件會讓方法稱為多載？
- 在寫程式的時候，如何判斷這個方法有多載？



文字檔處理入門

命名空間與類別

- `System.IO` 命名空間
- `File` 類別
 - <https://docs.microsoft.com/zh-tw/dotnet/api/system.io.file?redirectedfrom=MSDN&view=netframework-4.7.2>
- 這個類別類的方法都是 `static` (靜態方法)

我們將會用到的 `File` 類別中的方法

- `Exists`
- `ReadAllText`
- `ReadAllLines`
- `ReadLines`
- `WriteAllText`
- `WriteAllLines`
- `AppendAllText`
- `AppendAllLines`

LAB

閱讀文件

這個 Lab 不寫程式

- 請查詢 **Microsoft Docs** 文件庫，關於剛剛那六個方法（可能會有多載，請注意）
- 跟同學交互討論，輪流說明該方法的使用。

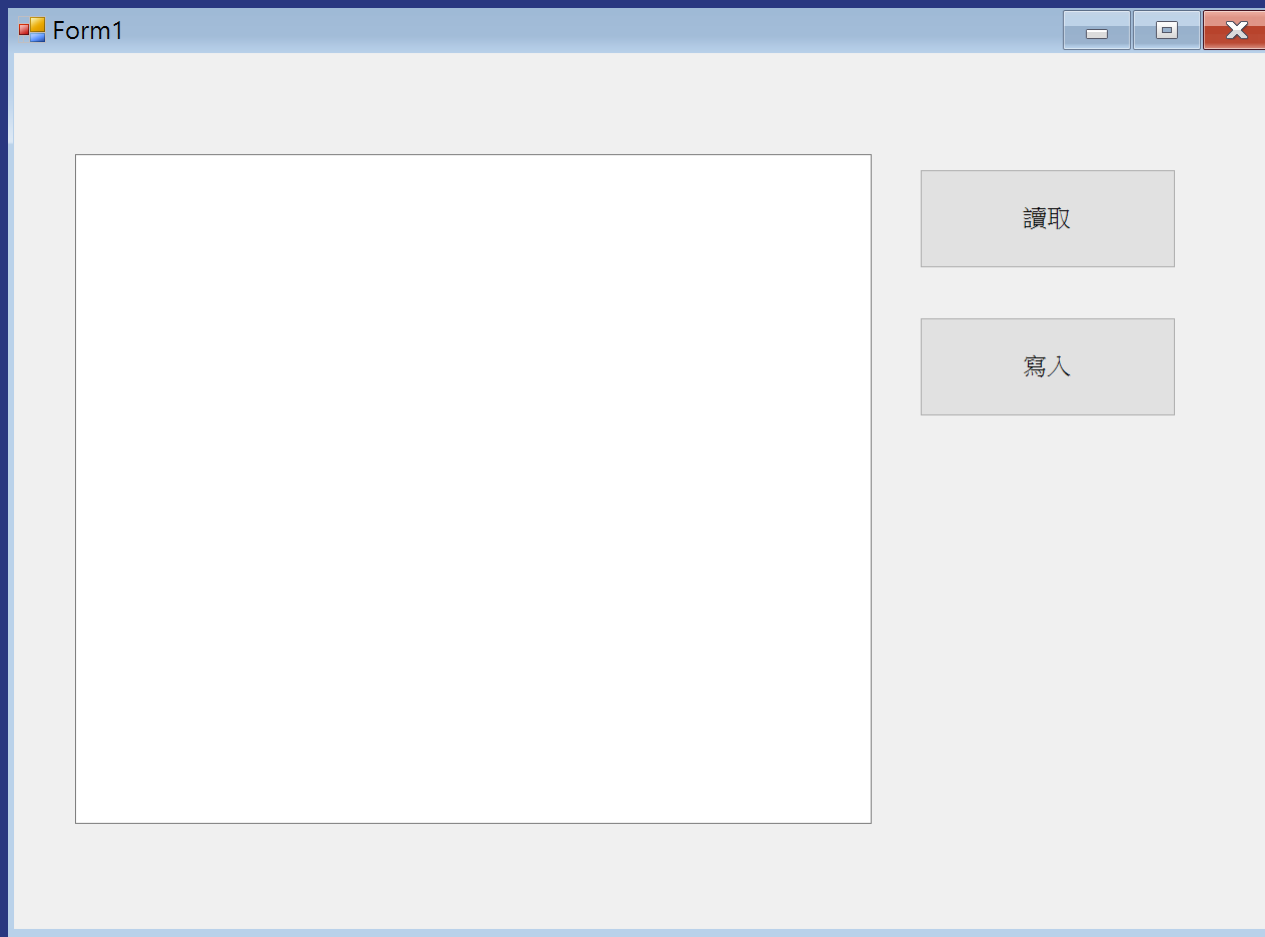
LAB

ReadAllText 與 WriteAllText

新增一個方案及專案

- 方案名稱：TextFileSamples
- 專案名稱：TextFileSample001
- 範本：Windows Forms Application

畫面配置



一個 TextBox
兩個 Button

TextBox 的 Multiline 屬性設定為 true
ScrollBars 屬性設定為 Vertical

在 Form1 Class 加入以下程式碼

```
public partial class Form1 : Form
{
    private string fileName;
    public Form1()
    {
        InitializeComponent();
        fileName = "test.txt";
    }

    private void button1_Click(object sender, EventArgs e)
    {
        textBox1.Clear();
        if (File.Exists(fileName))
        { textBox1.Text = File.ReadAllText(fileName); }
        else
        { MessageBox.Show("檔案不存在"); }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        File.WriteAllText(fileName, textBox1.Text);
        MessageBox.Show("存檔完成");
    }
}
```

執行

討論

- 為什麼在讀取檔案前要先呼叫 `File.Exists` ？如果拿掉他會發生甚麼事？
- 解釋整個程式的運作。



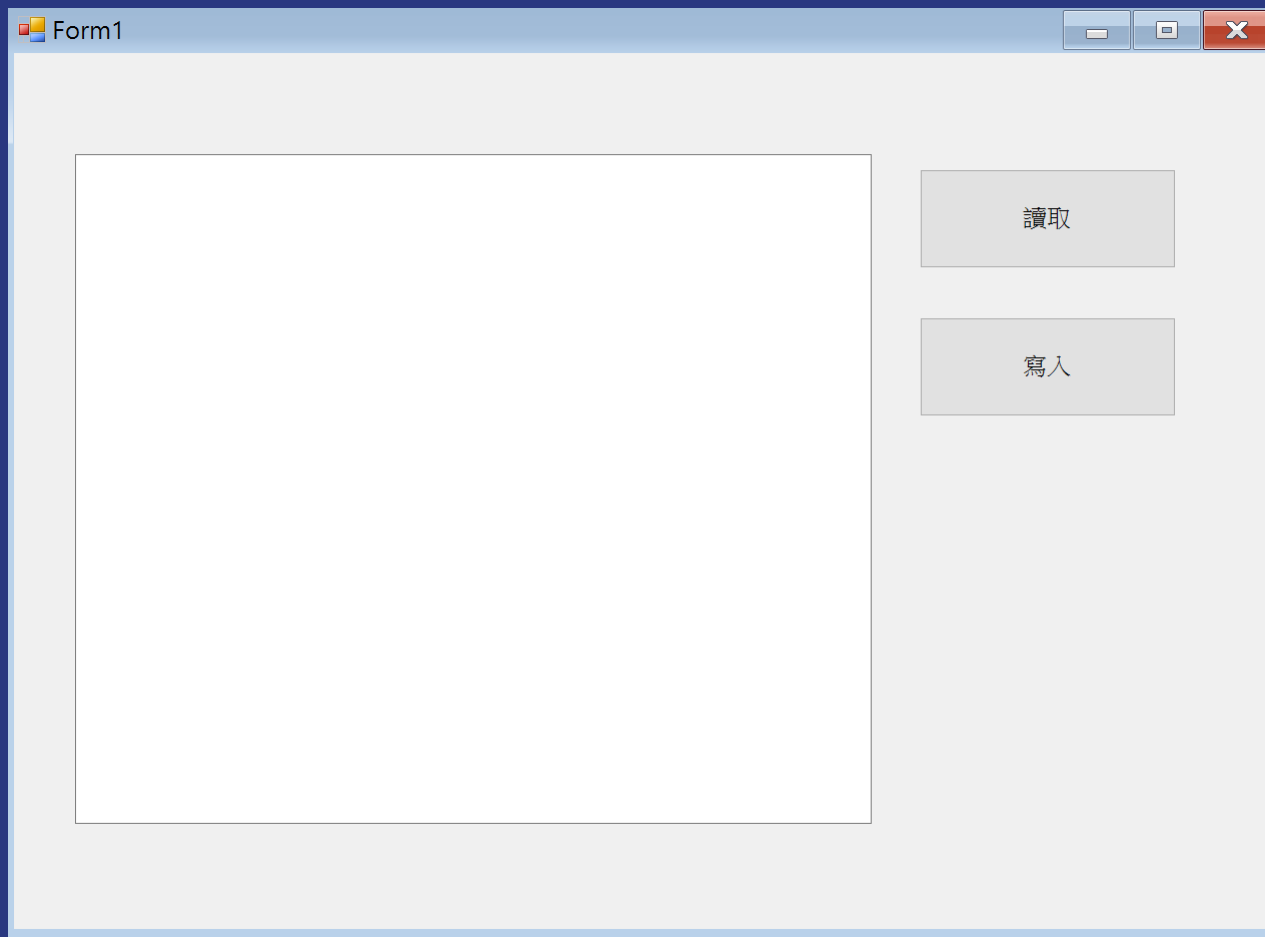
LAB

ReadAllLines

在 TextFileSamples 加入新專案

- 方案名稱： TextFileSamples
- 專案名稱： TextFileSamples002
- 範本：Windows Forms Application

畫面配置



一個 TextBox
兩個 Button

TextBox 的 Multiline 屬性設定為 true
ScrollBars 屬性設定為 Vertical

```
public partial class Form1 : Form
{
    private string fileName;
    public Form1()
    {
        InitializeComponent();
        fileName = "test.txt";
    }

    private void button1_Click(object sender, EventArgs e)
    {
        textBox1.Clear();
        if (File.Exists(fileName))
        {
            string[] lines = File.ReadAllLines(fileName);
            foreach (var line in lines)
            {
                textBox1.Text += line;
            }
        }
        else
        {
            MessageBox.Show("檔案不存在");
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        File.WriteAllText(fileName, textBox1.Text);
        MessageBox.Show("存檔完成");
    }
}
```

執行

討論

- ReadAllText 和 ReadAllLines 的不同



CSV File

- 逗號分隔值檔案，是一種利用逗號分隔資料的方式
- 也稱為字元分隔值，因為有些時候會用其他符號分隔
- 有些 CSV 的第一列是資料欄位名稱，第二列以後才是資料內容
- 有些 CSV 則全部都是資料內容
- 請開啟瀏覽器，連接到以下網頁
 - [原住民十大特色美食 | 政府資料開放平臺 \(data.gov.tw\)](https://data.gov.tw)

String.Split

- `String.Split` 方法可以依據傳入的特定字元分割字串
- [https://msdn.microsoft.com/zh-tw/library/system.string.split\(v=vs.110\).aspx](https://msdn.microsoft.com/zh-tw/library/system.string.split(v=vs.110).aspx)
- 請先閱讀上述文件。

DataGridView 控制項

- **DataGridView** 控制項以表格式顯示資料，是一項功能強大、有彈性的方式。您可以使用 **DataGridView** 控制來顯示少量資料的唯讀檢視，或者您可以調整它的大小，以顯示非常大的資料集之可編輯檢視。
- <https://docs.microsoft.com/zh-tw/dotnet/api/system.windows.forms.datagridview?redirectedfrom=MSDN&view=netframework-4.7.2>
- <https://docs.microsoft.com/zh-tw/dotnet/framework/winforms/controls/datagridview-control-windows-forms>

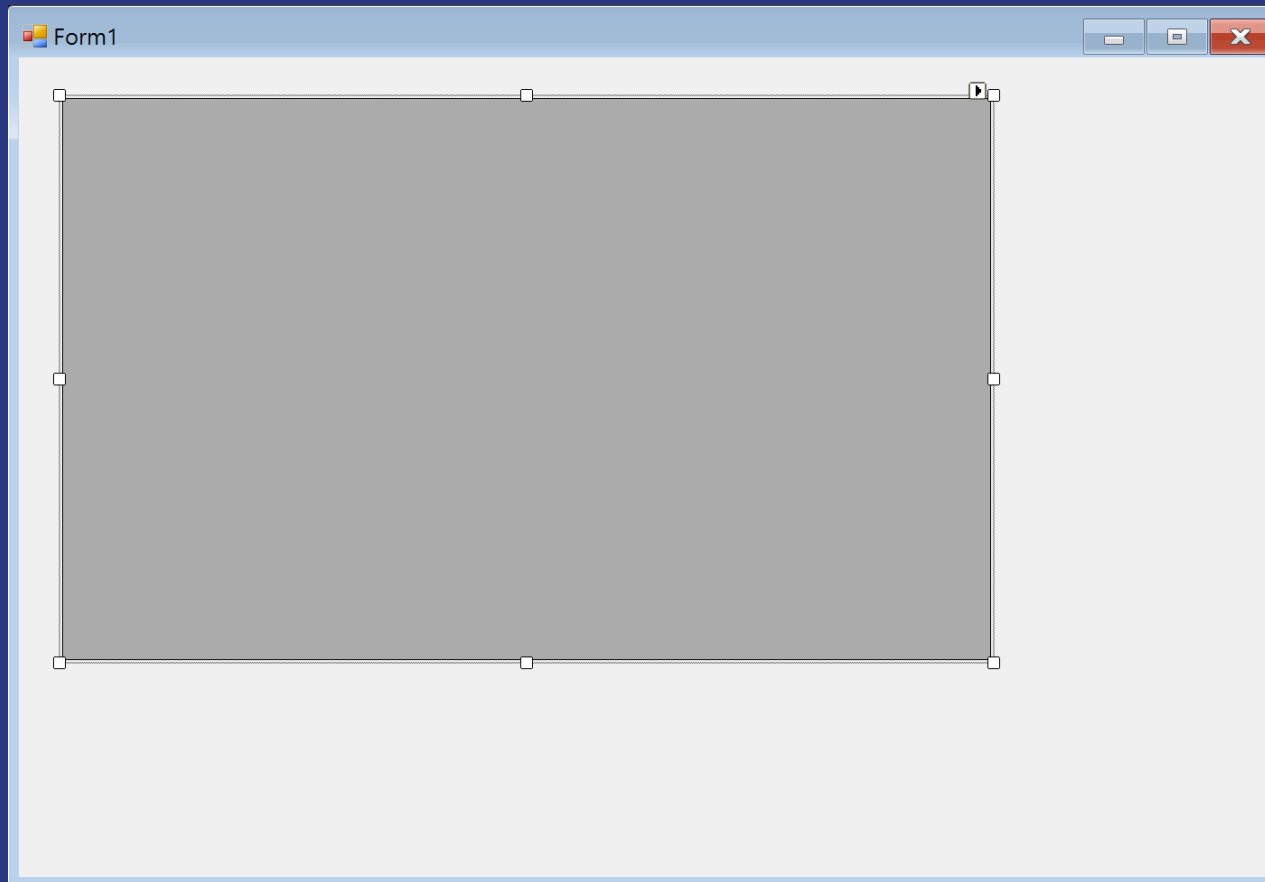
LAB

DataGridView 控制項

在 TextFileSamples 加入新專案

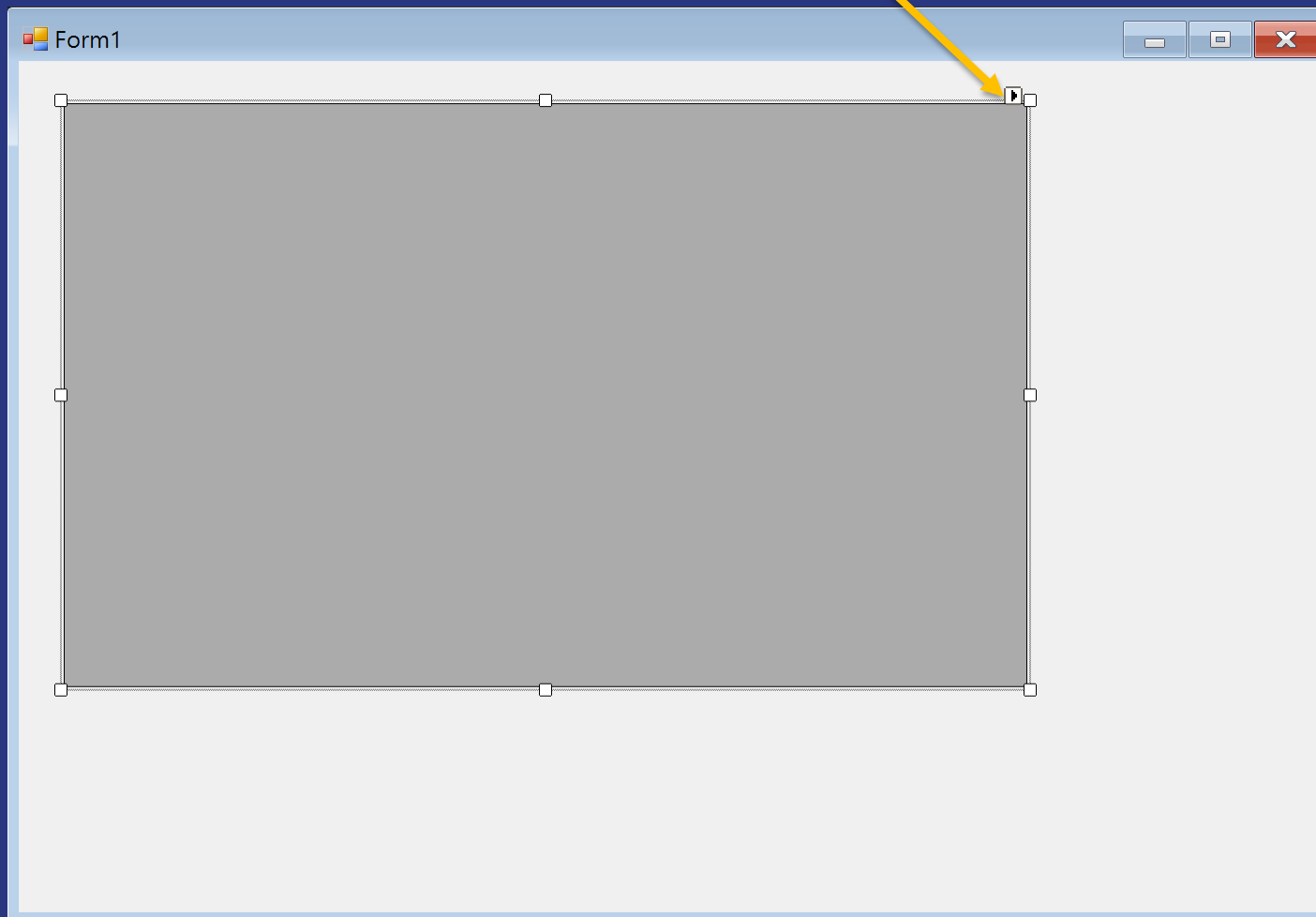
- 方案名稱： TextFileSamples
- 專案名稱： TextFileSamples003
- 範本：Windows Forms Application

畫面配置

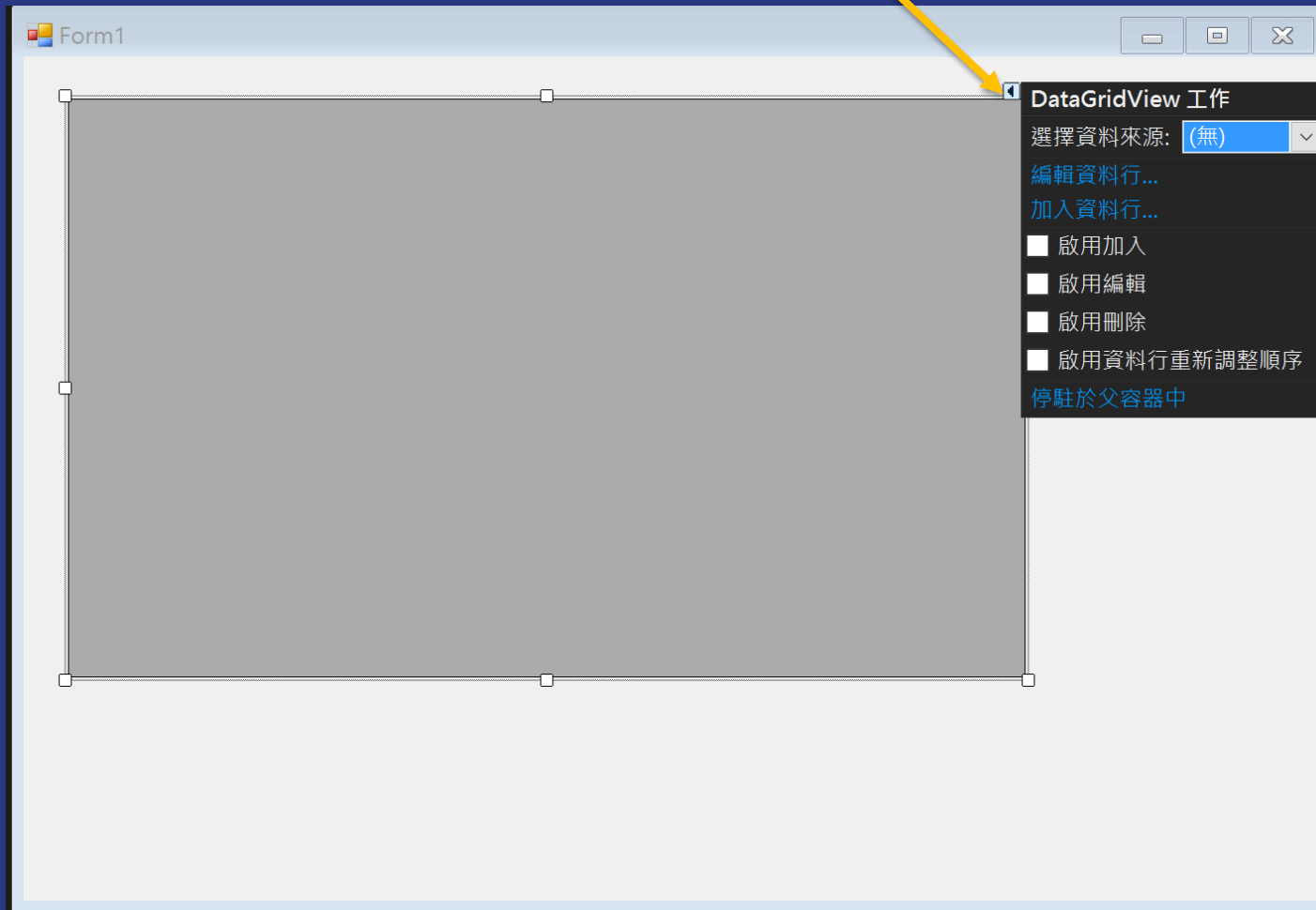


加入一個
DataGridView

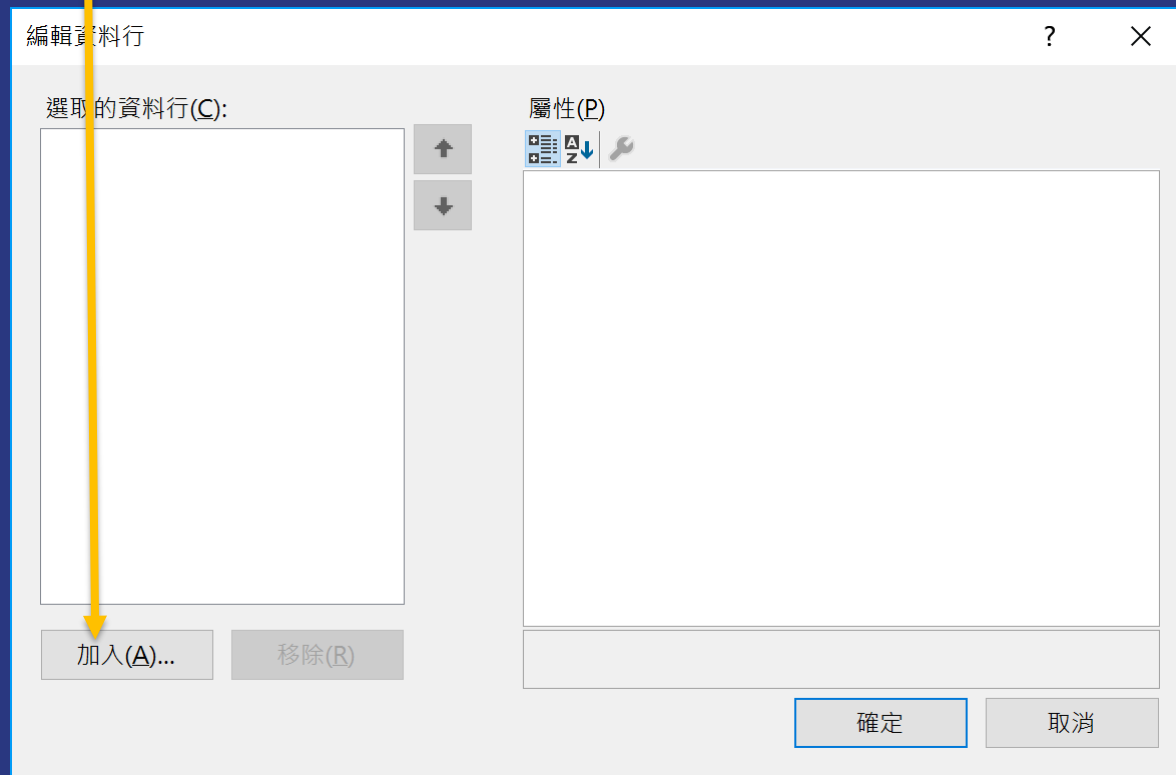
點選 DataGridView 上這個三角形



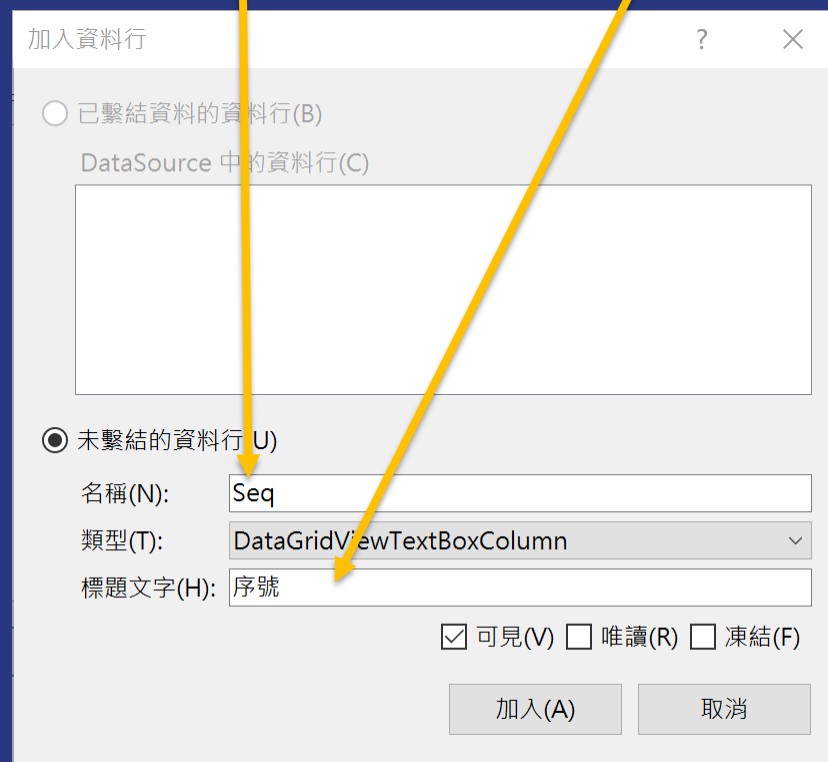
取消所有的勾選後，按下【編輯資料行】



按下【加入】



修改【名稱】為 Seq，【標題文字】為序號
按下【加入】



The screenshot shows a Windows-style dialog box titled '加入資料行' (Add Data Row). It has two radio buttons: '已繫結資料的資料行(B)' (Bound Data Row) and '未繫結的資料行(U)' (Unbound Data Row). The 'Unbound' option is selected. Below it is a section for 'DataSource 中的資料行(C)' (Data Row in DataSource) with an empty list box. Further down are three text input fields: '名稱(N):' (Name) containing 'Seq', '類型(T):' (Type) set to 'DataGridViewTextBoxColumn', and '標題文字(H):' (Header Text) containing '序號'. At the bottom right are three checkboxes: '可見(V)' (Visible) which is checked, '唯讀(R)' (Read-only), and '凍結(F)' (Frozen). At the very bottom are two buttons: '加入(A)' (Add) and '取消' (Cancel). Two yellow arrows point from the text above to the 'Seq' and '序號' fields.

加入資料行

☐ 已繫結資料的資料行(B)

DataSource 中的資料行(C)

☒ 未繫結的資料行(U)

名稱(N): Seq

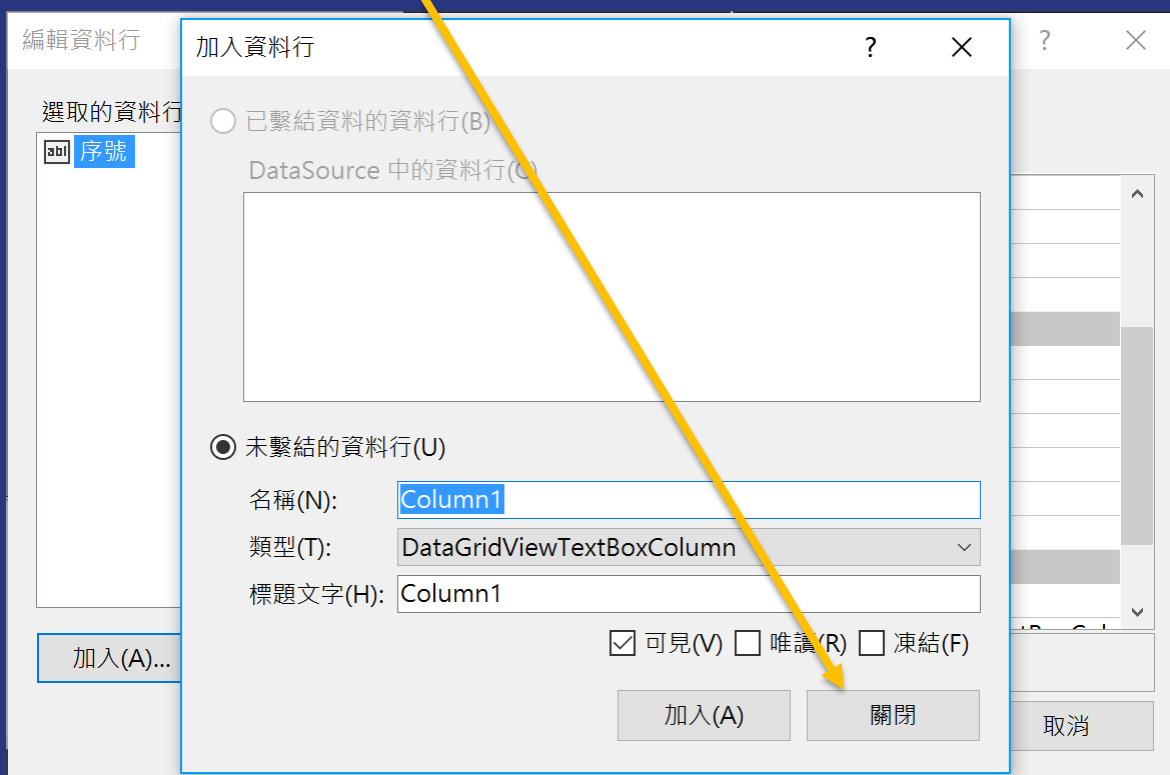
類型(T): DataGridViewTextBoxColumn

標題文字(H): 序號

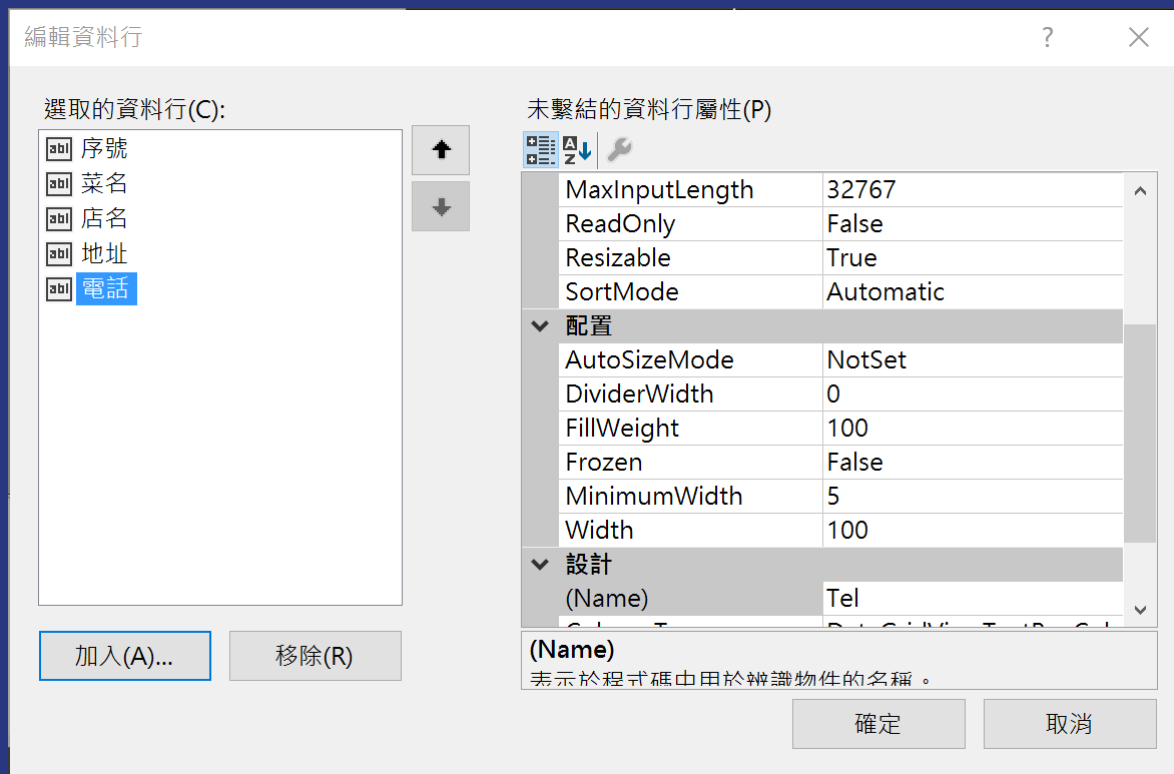
☒ 可見(V) ☐ 唯讀(R) ☐ 凍結(F)

加入(A) 取消

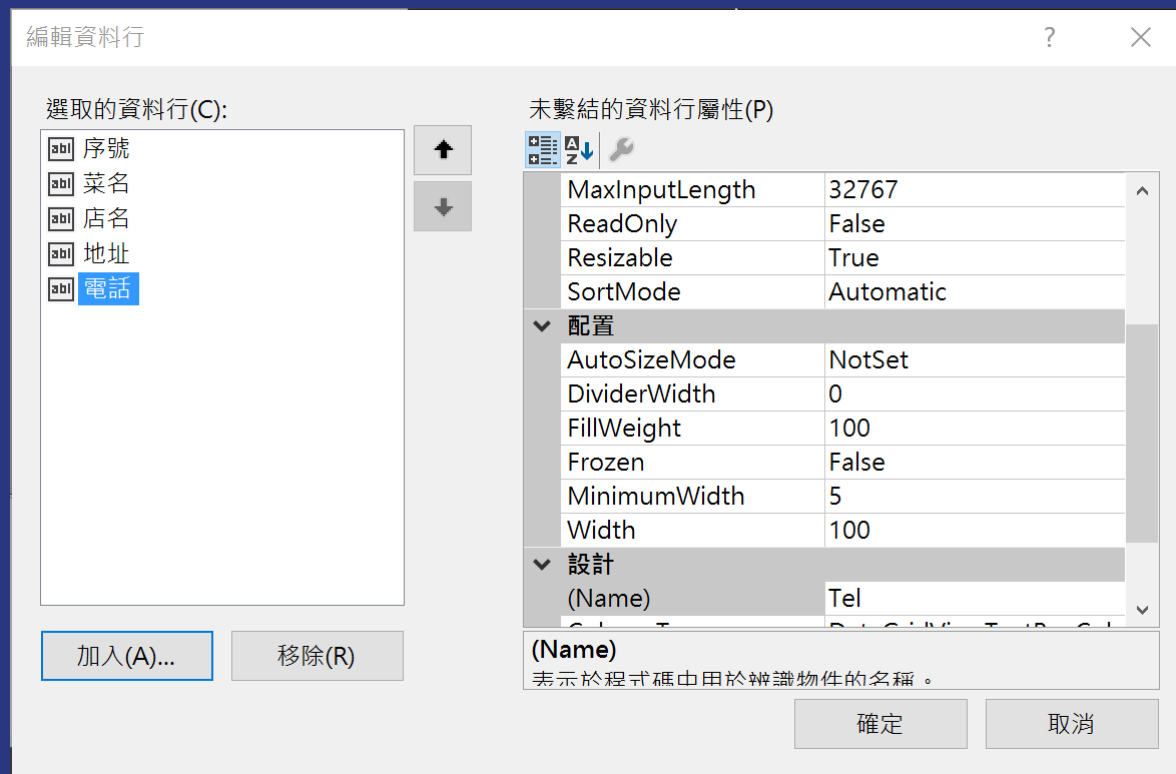
依序加入 DishName，菜名
Shop，店名
Address，地址
Tel，電話
完成後按【關閉】



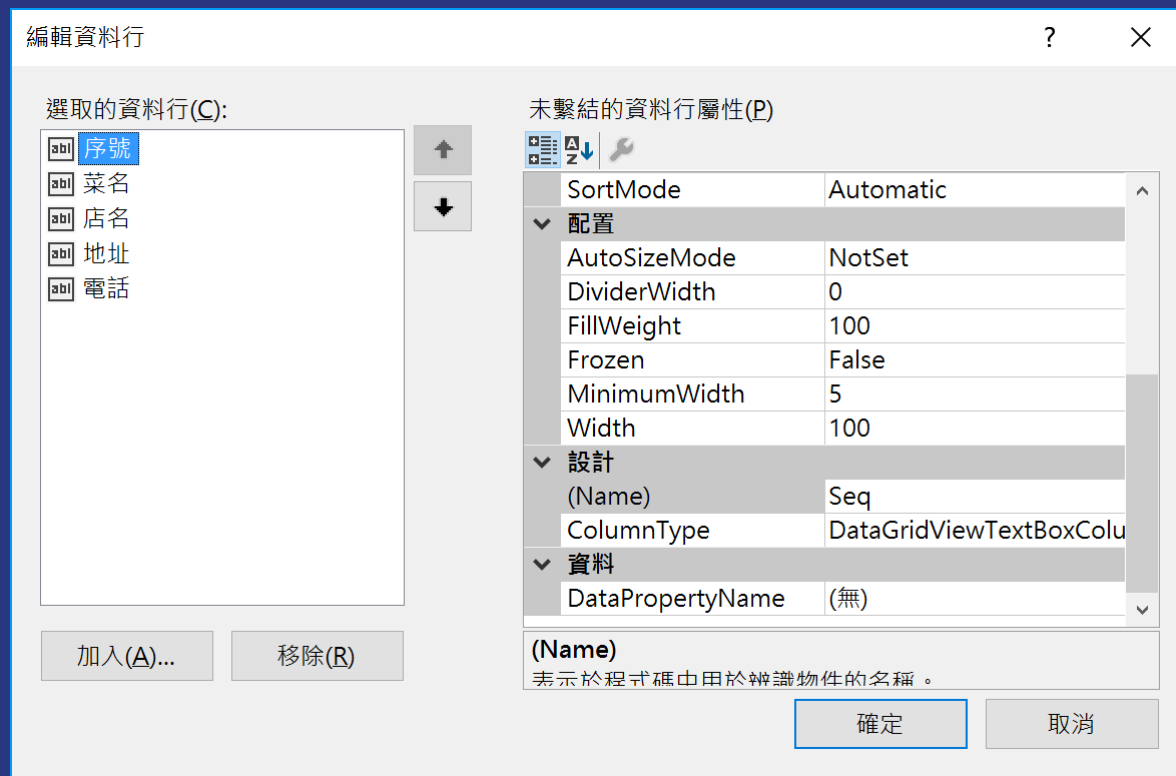
現在你應該看到這樣



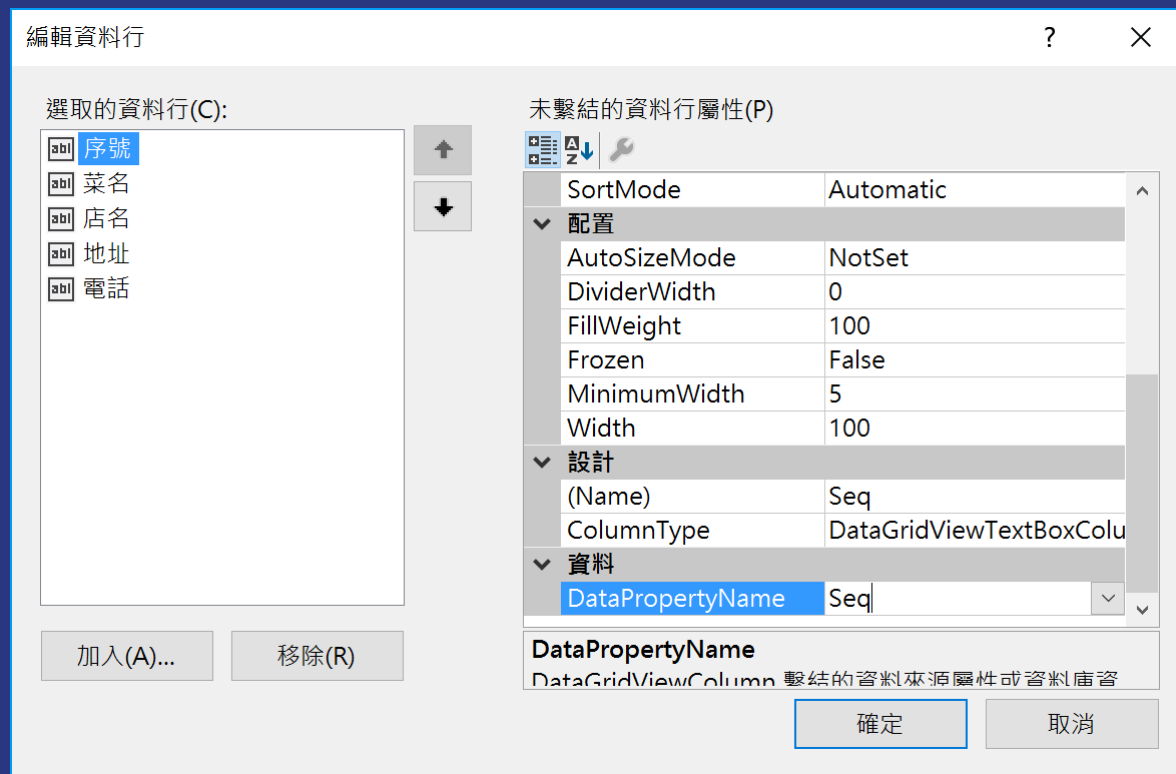
接著我們要設定資料欄位的對應



左邊先選【序號】，右邊捲軸拉到底
找到【DataPropertyName】



在【DataPropertyName】填入 Seq

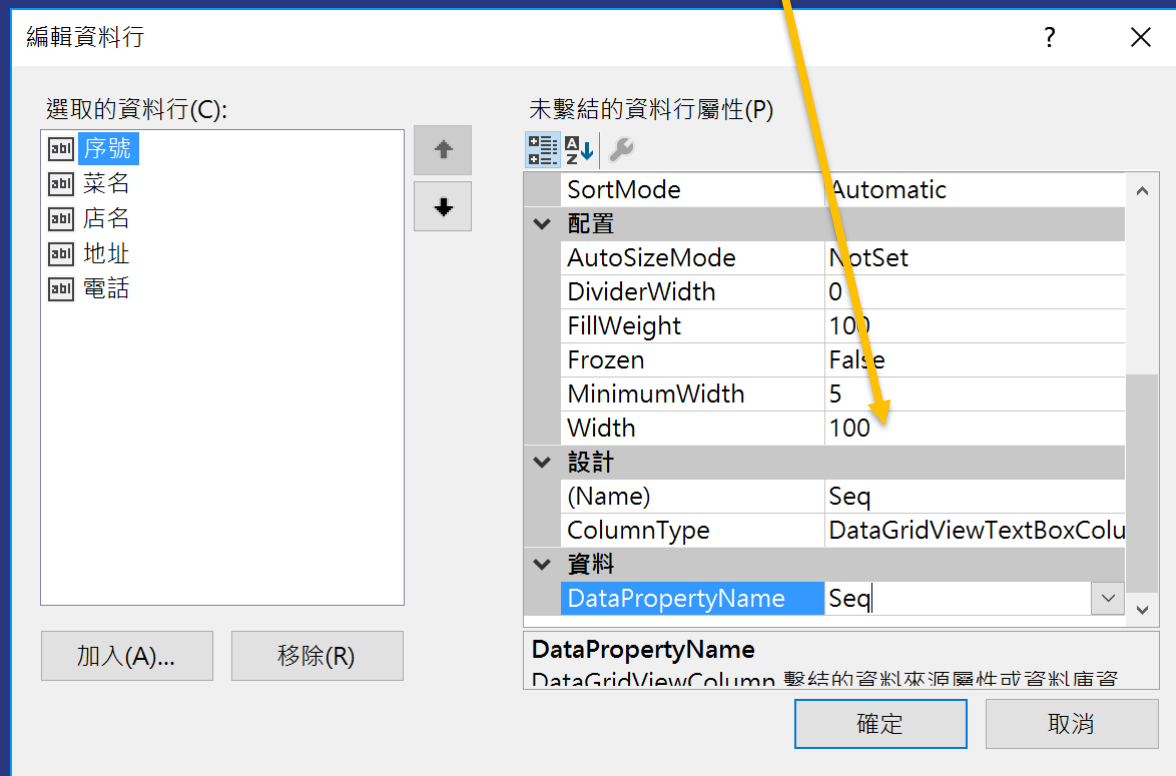


- 依序完成以下欄位對應
 - 菜名，DishName
 - 店名，Shop
 - 地址，Address
 - 電話，Tel
- 完成後按【確定】

Form1

序號	菜名	店名	地址	電話
----	----	----	----	----

回到【編輯資料行】找到【Width】屬性
調整每個欄的寬度



討論

- 回想剛剛整個對於 DataGridView 的設計操作，是否已經了解並能順利設計一個 DataGridView 了？



LAB

讀取寫入 CSV 檔案
(原住民十大特色美食)

繼續使用 TextSample003

- 下載這個 CSV 檔，並將它複製到這個專案的輸出目錄（ bin\debug）

加入 Restaurant Class

```
class Restaurant
{
    public int Seq { get; set; }

    public string DishName { get; set; }

    public string Shop { get; set; }

    public string Address { get; set; }

    public string Tel { get; set; }
}
```

在 Form1 Class 建立取得資料的 Method

```
private List<Restaurant> CreateData()
{
    string fileName = "102015.csv";
    if (File.Exists(fileName))
    {
        return File.ReadLines(fileName).Skip(1)
            .Select(x => x.Split(','))
            .Select(x => new Restaurant
            {
                Seq = int.Parse(x[0]),
                DishName = x[1],
                Shop = x[3],
                Address = x[4],
                Tel = x[5]
            }).ToList();
    }
    return null;
}
```

修改 Form1 Class 的建構式

```
public Form1()  
{  
    InitializeComponent();  
    var list = CreateData();  
    dataGridView1.DataSource = list;  
}
```

執行

JSON File

- JSON (JavaScript Object Notation) 是一種輕量級的資料交換語言，以文字為基礎，且易於讓人閱讀。
- 參考 <https://zh.wikipedia.org/wiki/JSON>
- 請開啟瀏覽器，連接到以下網頁
 - <http://data.gov.tw/dataset/6083>

NuGet

- NuGet 是一種 Visual Studio 的擴充
- 讓我們可以快速地引用別人寫好的函式庫
- 例如 : Newtonsoft.Json (俗稱 Json.Net)

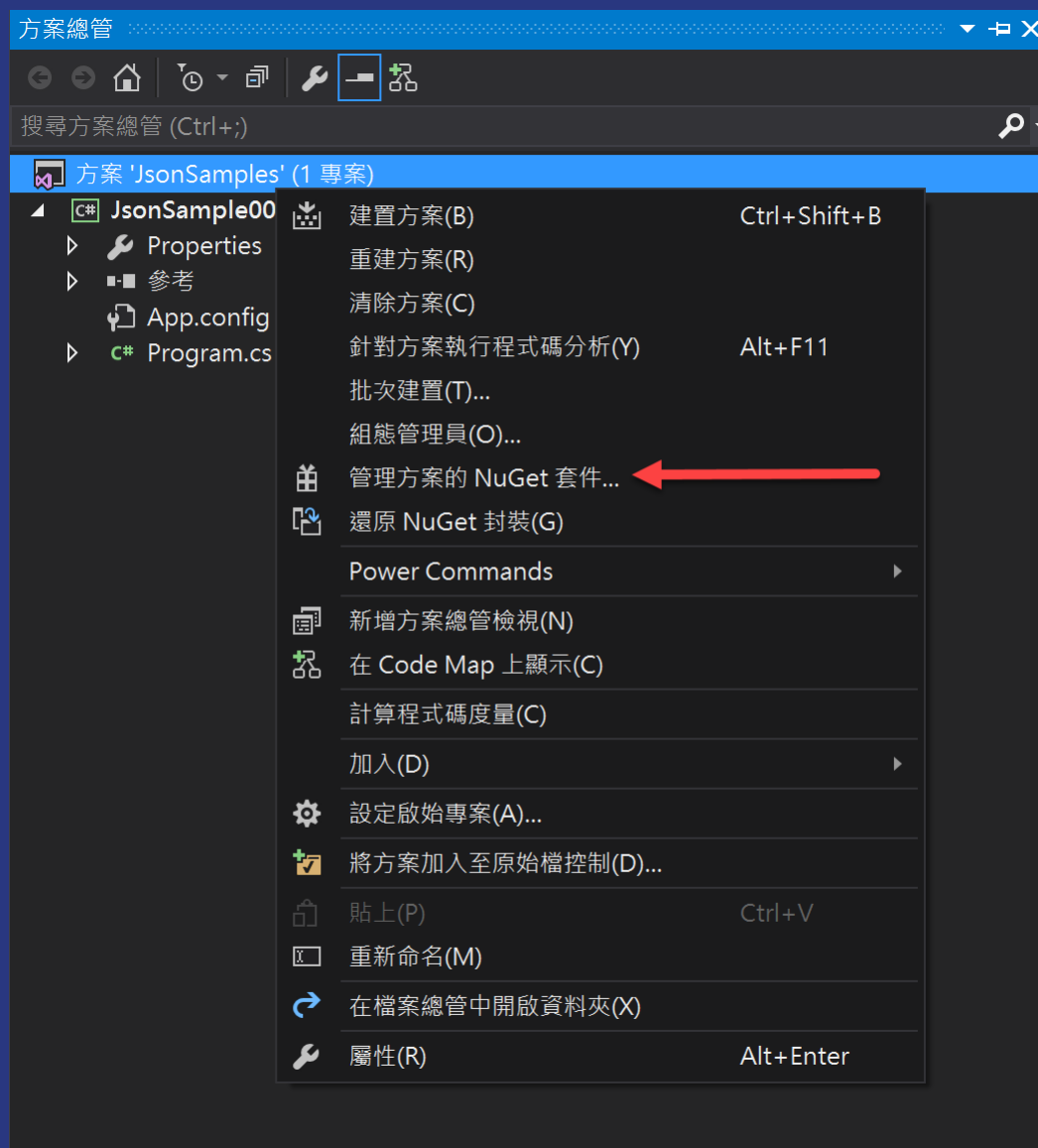
LAB

使用 NuGet
取得 Newtonsoft.Json 套件

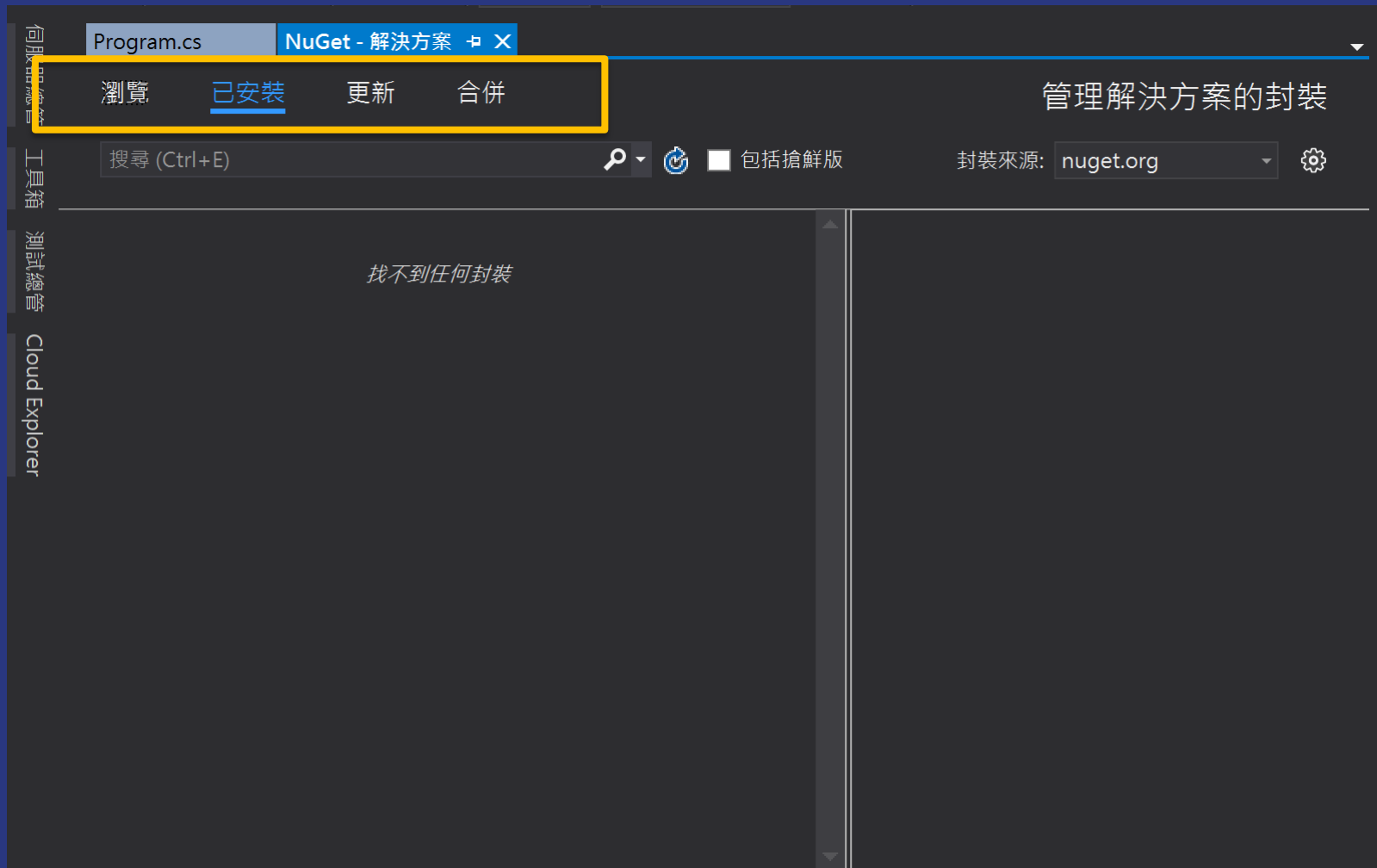
新增一個方案及專案

- 方案名稱：JsonSamples
- 專案名稱：JsonSample001
- 範本：Windows Forms Application

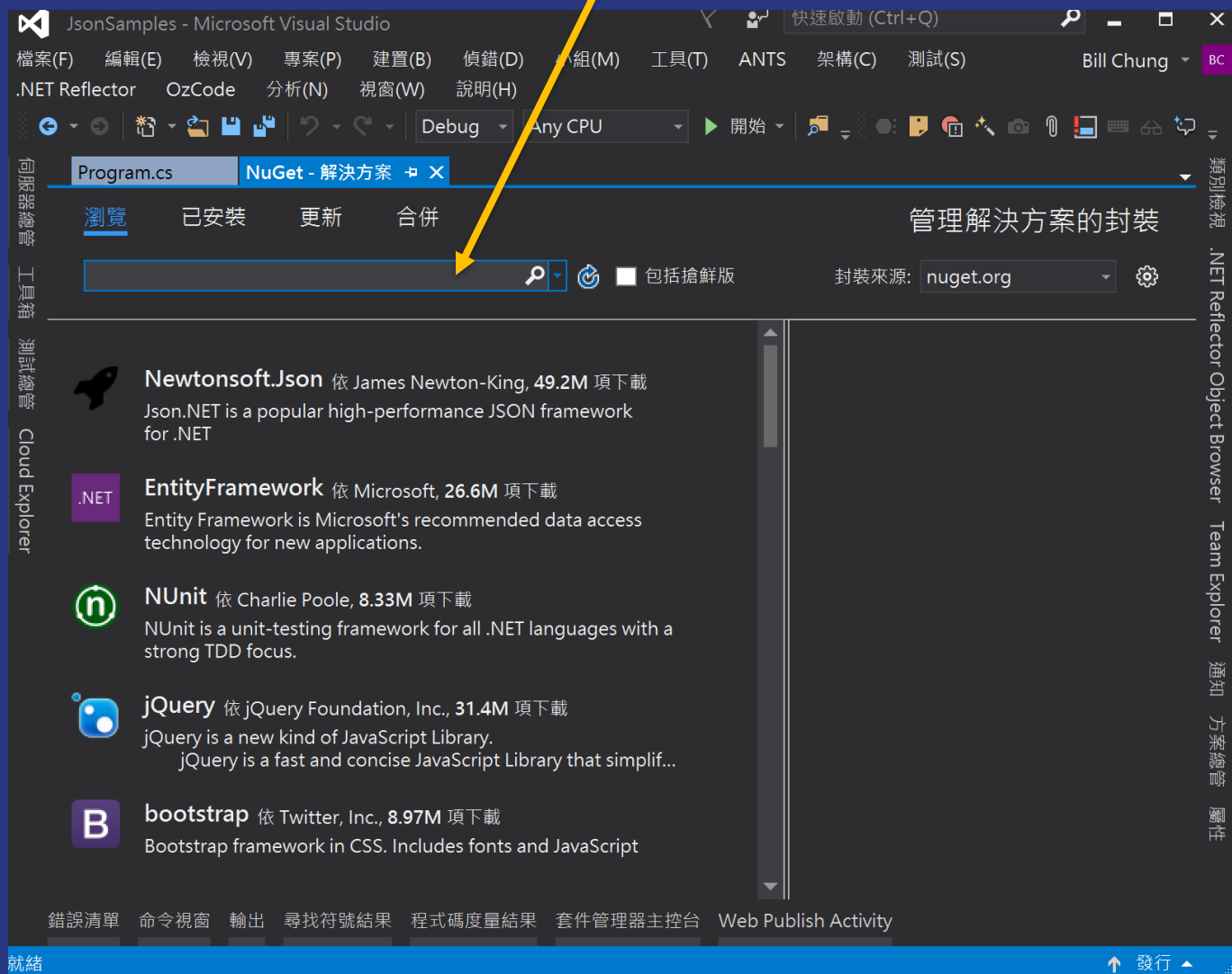
在方案總管中，選擇方案，然後按下滑鼠右鍵會看到【管理方案的 NuGet 套件】



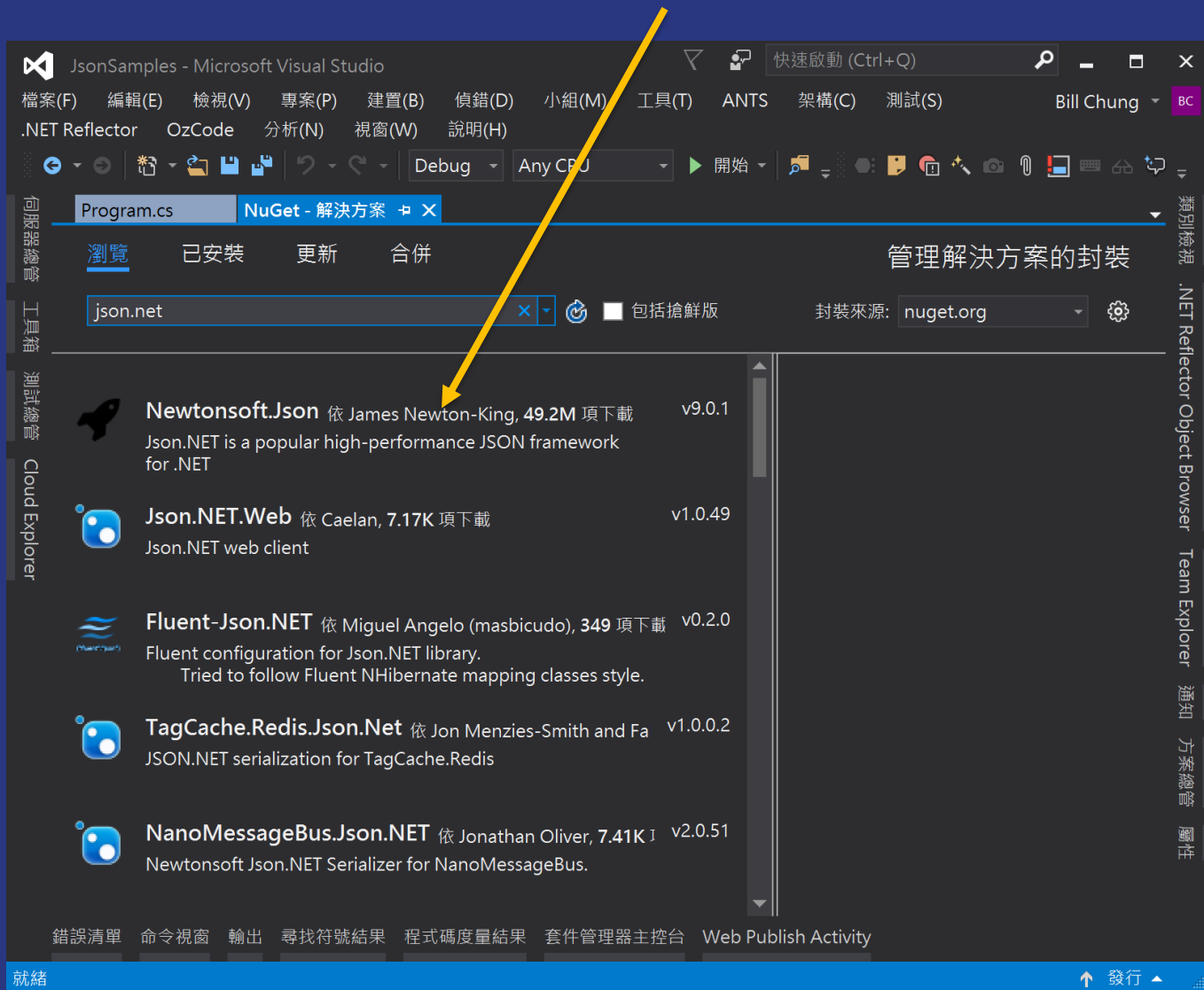
我們現在要去 NuGet 找我們需要的套件
所以請點選【瀏覽】



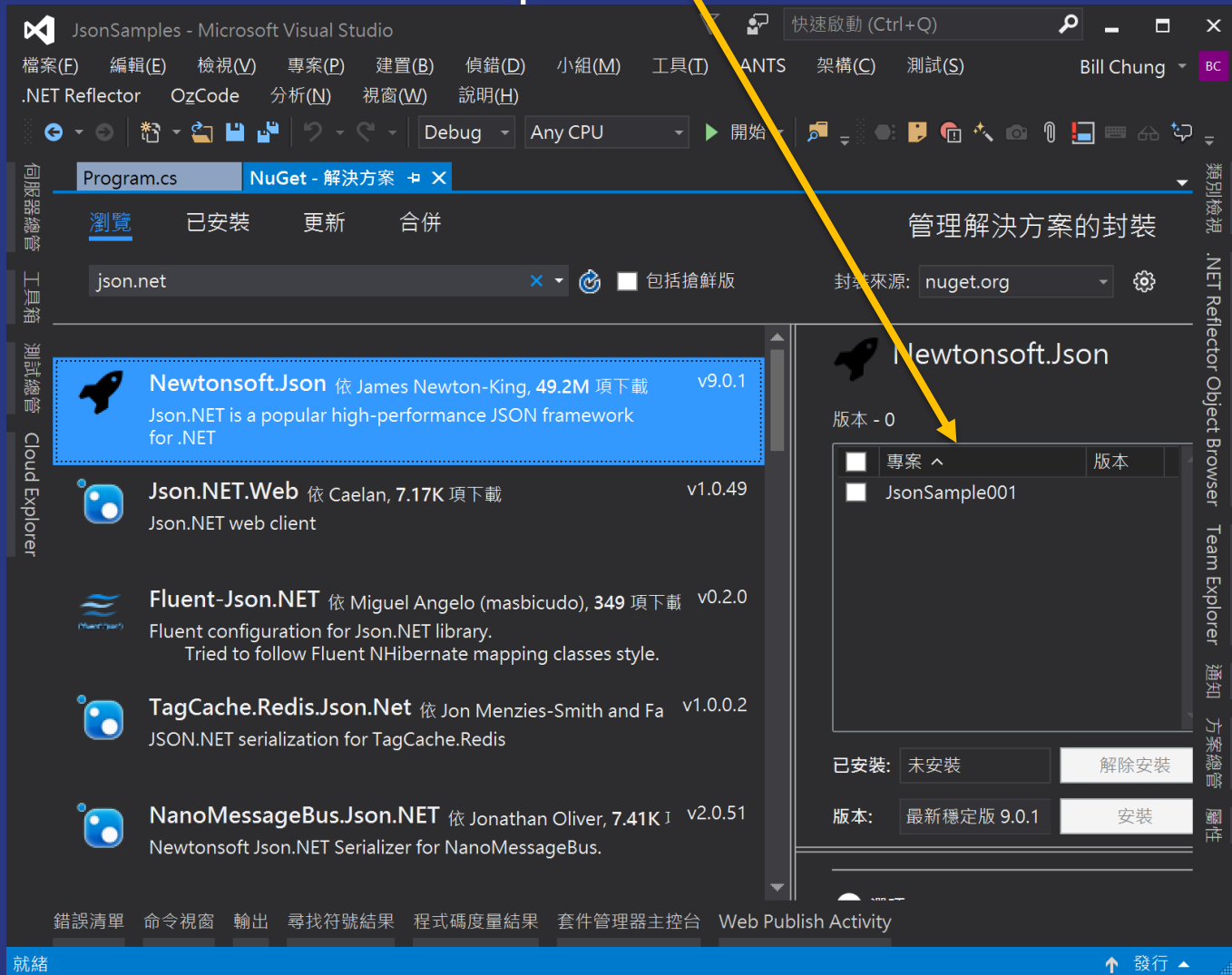
雖然第一個就是我們要用的，但你不會常常運氣這麼好
所以請在瀏覽列中輸入 “Json.Net” 或 "Newtonsoft.Json"



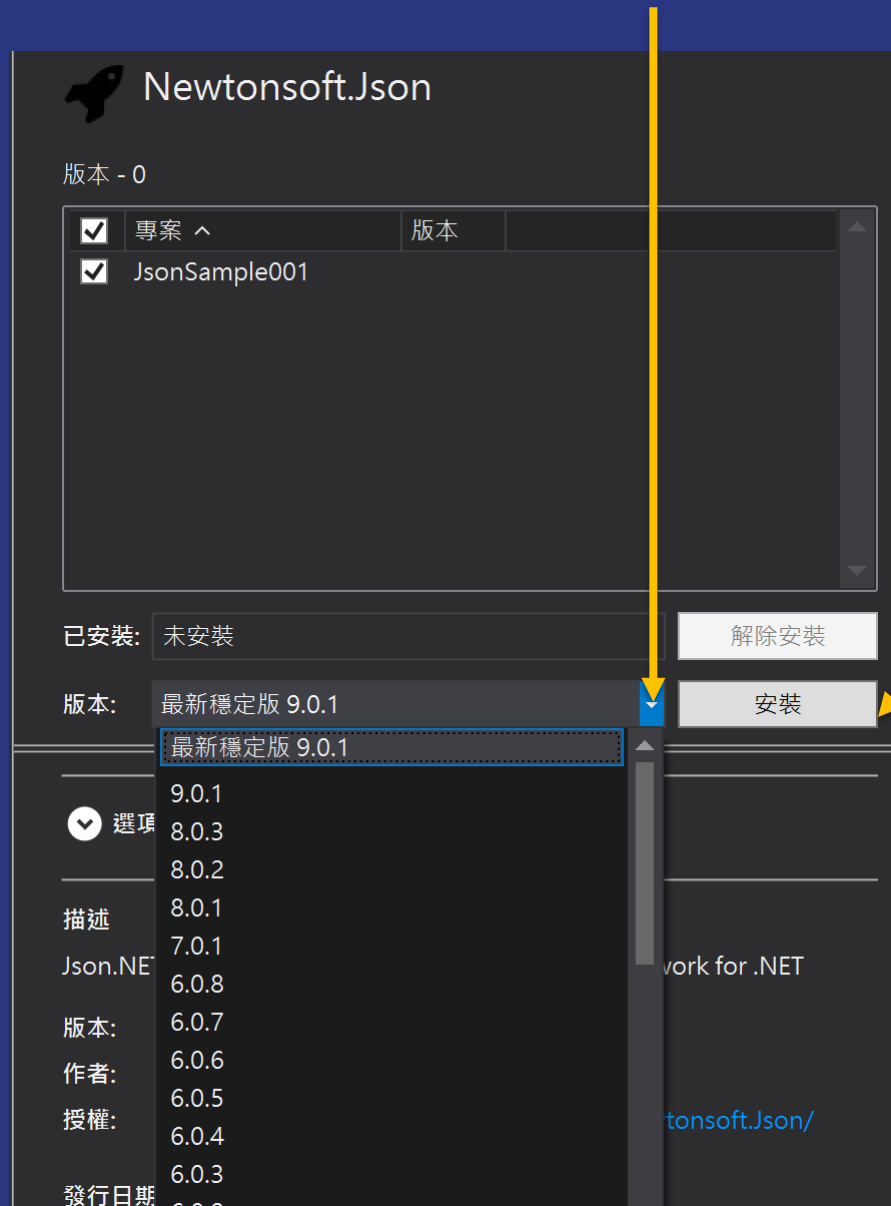
會看到一堆相關的套件，我們要用的是第一個
所以選取它



右邊窗格代表你要在這個方案中的哪些專案
安裝或解除安裝這個套件
請勾選 【JsonSample001】

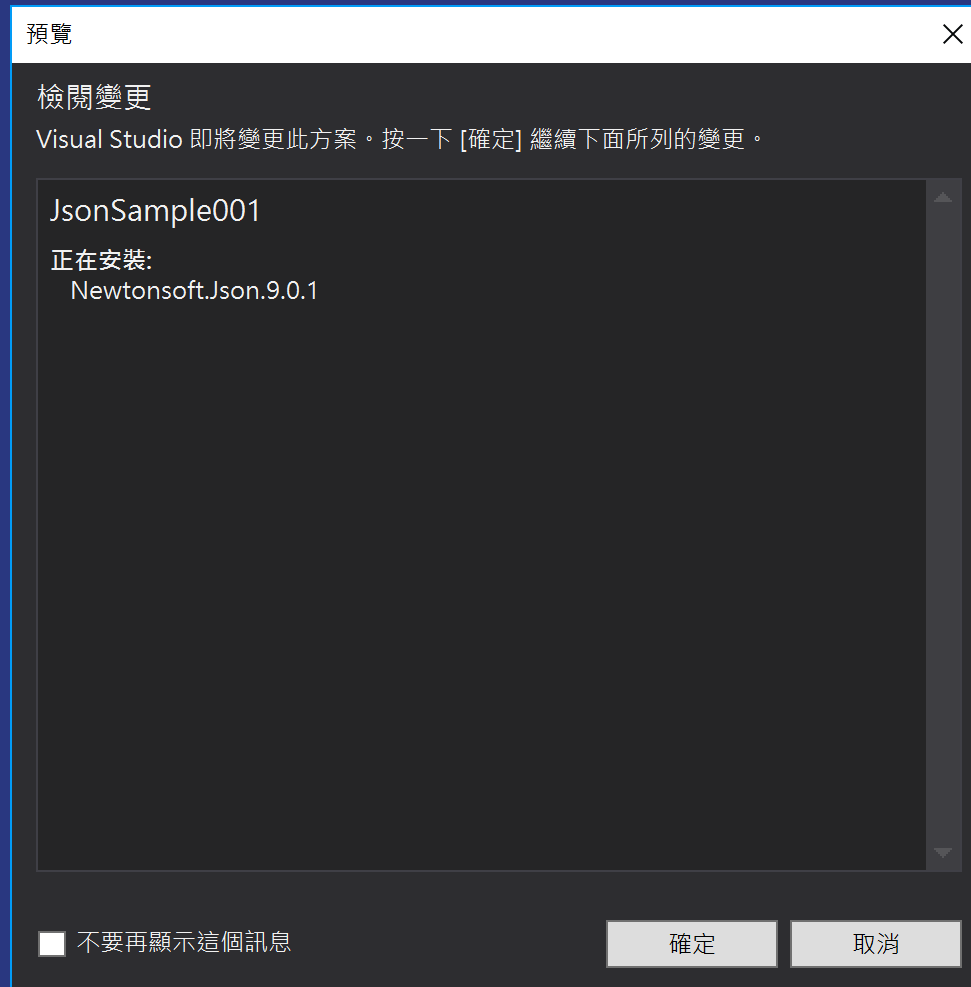


(1) 這邊可以選擇套件的版本，通常選『最新穩定版』



(2) 然後按下【安裝】

出現任何提示畫面都要注意它的內容
此處告訴你，他將要安裝到 JsonSample001 專案裡
請按【確定】



安裝過程注意【輸出】視窗告訴你的訊息

The screenshot shows the Visual Studio Package Manager interface. The top bar includes tabs for '瀏覽' (Browse), '已安裝' (Installed), '更新' (Updates), and '合併' (Merge). The search bar contains 'json.net'. The package list on the left shows several JSON-related packages, with 'Newtonsoft.Json' selected. The right pane displays details for 'Newtonsoft.Json' version 9.0.1, including a table of installed versions and a '解除安裝' (Uninstall) button. The bottom pane shows the '輸出' (Output) window with the '顯示輸出來源(S): 封裝管理員' (Show output from: Package Manager) selected. The output text provides a detailed log of the installation process, including dependency resolution, package download, and successful installation.

管理解決方案的封裝

封裝來源: nuget.org

Newtonsoft.Json

版本 - 1

專案 ^	版本
JsonSample001	9.0.1

已安裝: 9.0.1 解除安裝

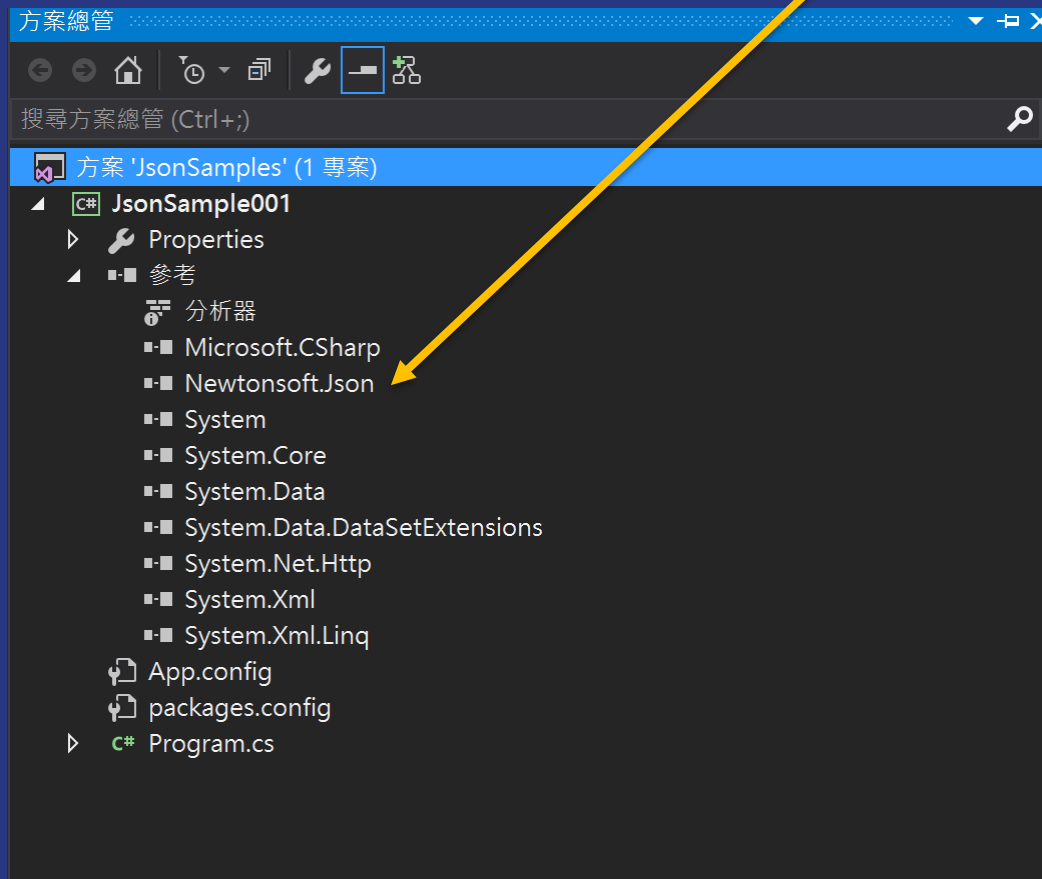
版本: 最新穩定版 9.0.1 安裝

輸出

顯示輸出來源(S): 封裝管理員

嘗試解析具有 DependencyBehavior 'Lowest' 之封裝 'Newtonsoft.Json.9.0.1' 的相依性
解析相依性資訊花費了 0 ms
正在解析安裝封裝 'Newtonsoft.Json.9.0.1' 的動作
已解析安裝封裝 'Newtonsoft.Json.9.0.1' 的動作
正在從 'nuget.org' 擷取封裝 'Newtonsoft.Json.9.0.1'。
正在將封裝 'Newtonsoft.Json.9.0.1' 加入資料夾 'C:\CUST_Samples\6\JsonSamples\packages'
已將封裝 'Newtonsoft.Json.9.0.1' 加入資料夾 'C:\CUST_Samples\6\JsonSamples\packages'
已將封裝 'Newtonsoft.Json.9.0.1' 加入 'packages.config'
正在執行指令檔 'C:\CUST_Samples\6\JsonSamples\packages\Newtonsoft.Json.9.0.1\tools\install.ps1'...
已成功將 'Newtonsoft.Json.9.0.1' 安裝到 JsonSample001
執行 NuGet 動作花費了 2.92 sec
已完成
經過時間: 00:01:45.3795274

安裝完成後，開啟方案總管，點開【參考】
會發現多了一個組件【Newtonsoft.Json】



JSON 的序列化與反序列化

- 所謂序列化，就是將我們在 C# 裡的建立的物件依照某種特定格式轉換成可保存或傳輸之形式的程序。
 - JSON 序列化就是將物件轉換成 JSON 的格式
 - 最簡單的入門就是用 `Newtonsoft.Json` 中的 `JsonConvert.SerializeObject` 方法
- 反序列化的作用則相反，會將 JSON 資料轉換回C#的物件
 - 使用 `JsonConvert.DeserializeObject<T>` 方法

關於 Json.Net 的基礎相關資料

- <https://dotblogs.com.tw/larrynung/2012/04/05/71295>
- <http://kevintsengtw.blogspot.com/2017/06/visual-studio-2017-json-c.html>
- 或請善用搜尋引擎

LAB

利用資料來源直接建立對應的類別

回到公開資料平台（原住民十大特色美食）

- <http://data.gov.tw/dataset/6083>

原住民十大特色美食



檢視

活化應用清單

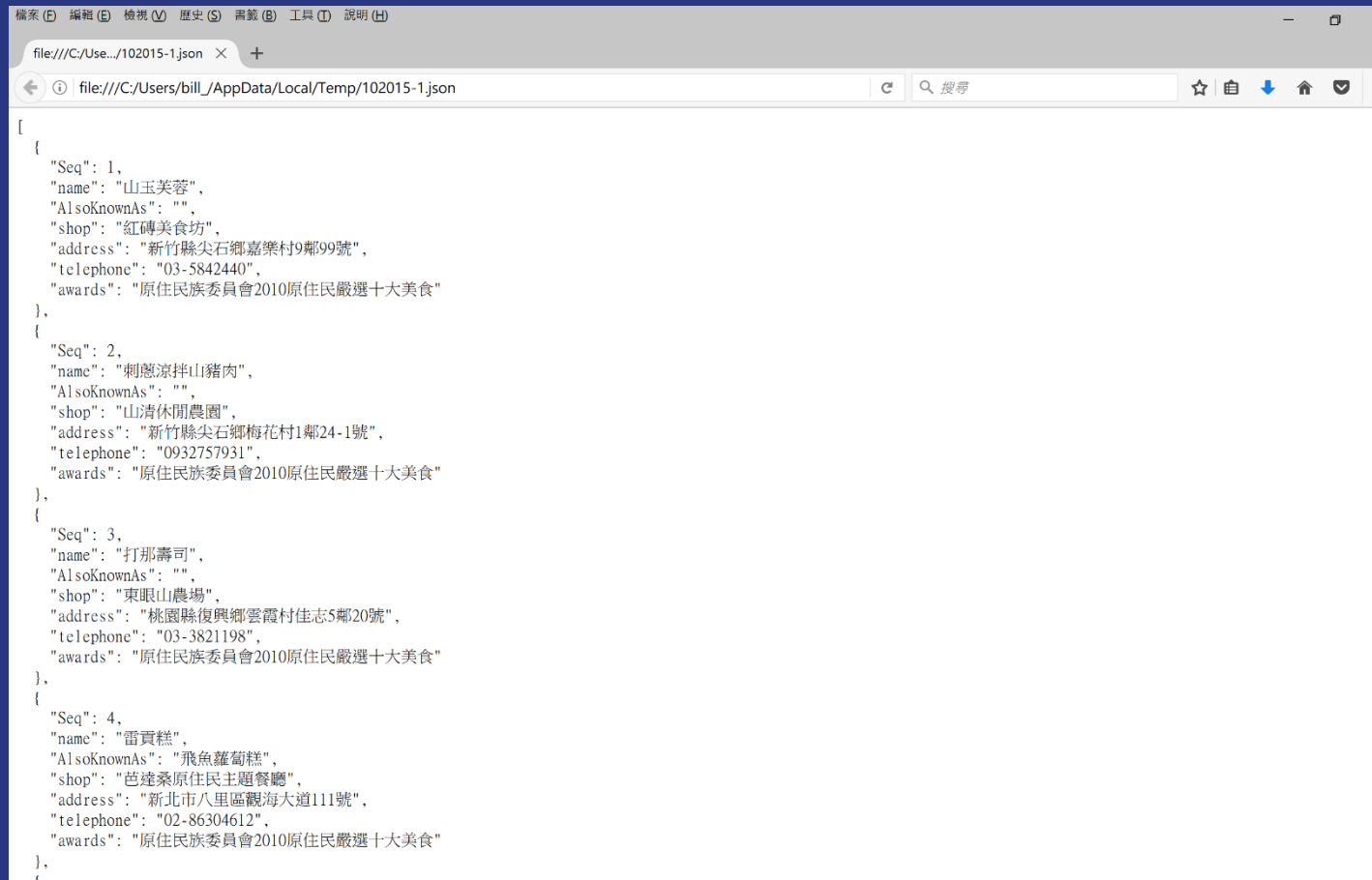
資料集評分:



平均: 4.8 (4 投票)

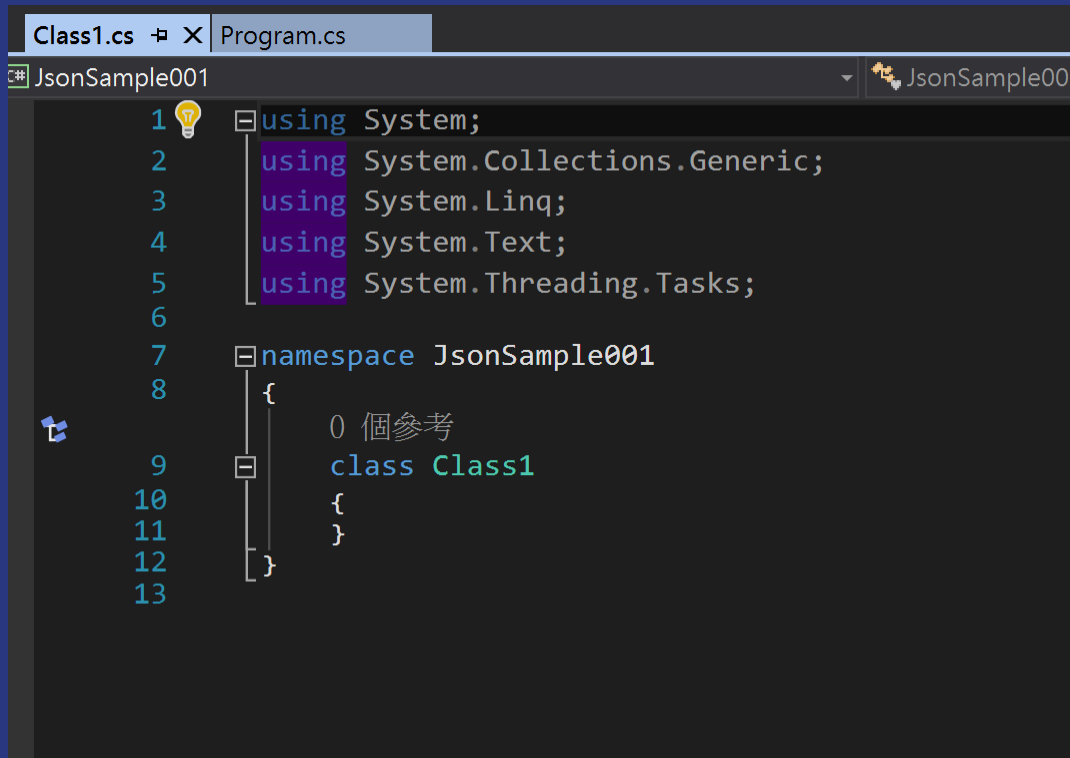
資料集描述	依行政院原住民族委員會辦理之「2010原住民嚴選十大獻禮」甄選活動，評選出訪各地原鄉時享原味之參考。	
主要欄位說明	項次、美食名稱、店名、地址、電話	
資料資源	 XML	 檢視資料
	 JSON	 檢視資料
	 CSV	 檢視資料
資料集類型	原始資料	

不論用任何方法，總之你會看到資料內容
先把資料【全選】→【複製】



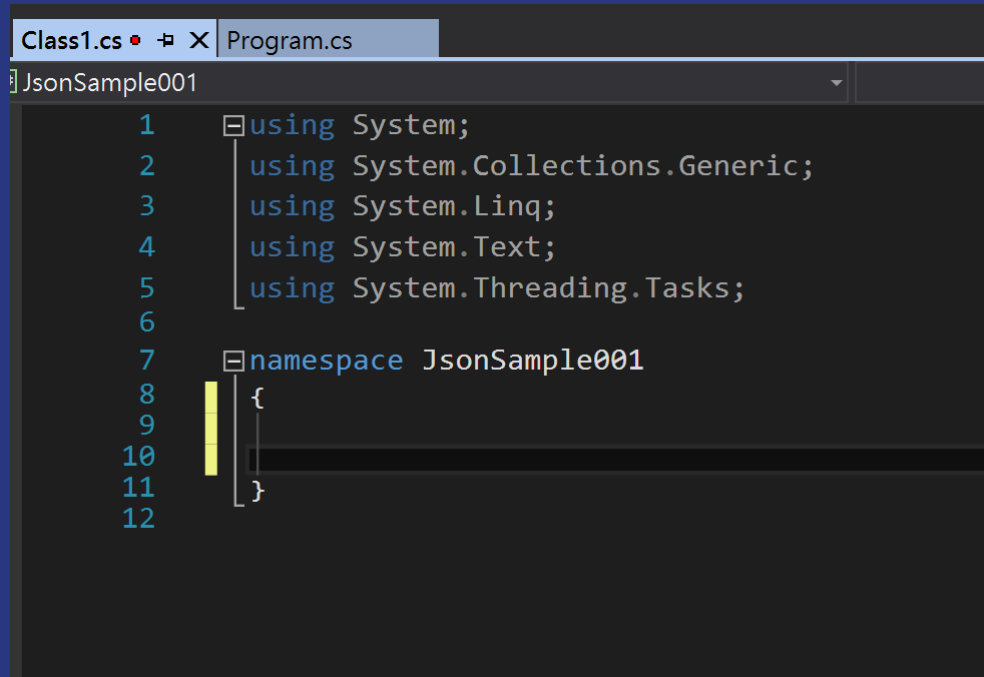
```
[
  {
    "Seq": 1,
    "name": "山玉芙蓉",
    "AlsoKnownAs": "",
    "shop": "紅磚美食坊",
    "address": "新竹縣尖石鄉嘉樂村9鄰99號",
    "telephone": "03-5842440",
    "awards": "原住民族委員會2010原住民嚴選十大美食"
  },
  {
    "Seq": 2,
    "name": "刺蔥涼拌山豬肉",
    "AlsoKnownAs": "",
    "shop": "山清休閒農園",
    "address": "新竹縣尖石鄉梅花村1鄰24-1號",
    "telephone": "0932757931",
    "awards": "原住民族委員會2010原住民嚴選十大美食"
  },
  {
    "Seq": 3,
    "name": "打那壽司",
    "AlsoKnownAs": "",
    "shop": "東眼山農場",
    "address": "桃園縣復興鄉雲霞村佳志5鄰20號",
    "telephone": "03-3821198",
    "awards": "原住民族委員會2010原住民嚴選十大美食"
  },
  {
    "Seq": 4,
    "name": "雷貢糕",
    "AlsoKnownAs": "飛魚蘿蔔糕",
    "shop": "芭達桑原住民主題餐廳",
    "address": "新北市八里區觀海大道111號",
    "telephone": "02-86304612",
    "awards": "原住民族委員會2010原住民嚴選十大美食"
  }
]
```

回到你的程式碼，隨便新增個什麼類別檔案
然後把 class Class1 刪除



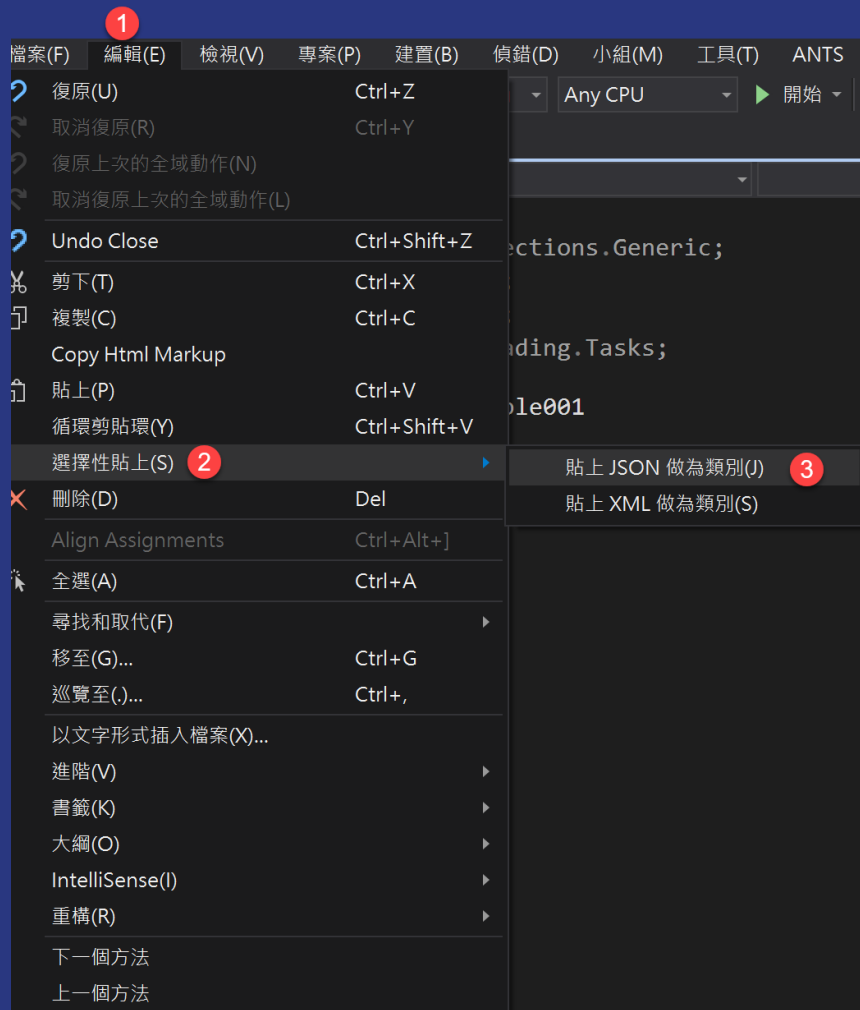
```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace JsonSample001
8 {
9     class Class1
10     {
11     }
12 }
13
```

刪除後讓游標保持在 namespace 區段內
準備『一秒變格格』



```
Class1.cs x Program.cs
JsonSample001
1  using System;
2      using System.Collections.Generic;
3      using System.Linq;
4      using System.Text;
5      using System.Threading.Tasks;
6
7  namespace JsonSample001
8  {
9
10
11  }
12
```

【編輯】→【選擇性貼上】→
【貼上 JSON 做為類別】
如果不能選，請回去把資料再度複製一遍



Visual Studio 自動依據資料內容產出必要的類別 (通常我們會修改這些類別的名稱)

```
{
    0 個參考
    public class Rootobject
    {
        0 個參考
        public Class1[] Property1 { get; set; }
    }

    1 個參考
    public class Class1
    {
        0 個參考
        public bool success { get; set; }
        0 個參考
        public Result result { get; set; }
    }

    1 個參考
    public class Result
    {
        0 個參考
        public string resource_id { get; set; }
        0 個參考
        public Field[] fields { get; set; }
        0 個參考
        public Record[] records { get; set; }
        0 個參考
        public int limit { get; set; }
        0 個參考
        public int offset { get; set; }
        0 個參考
        public int total { get; set; }
    }
    1 個參考
```

- 先把 Rootobject class 註解掉

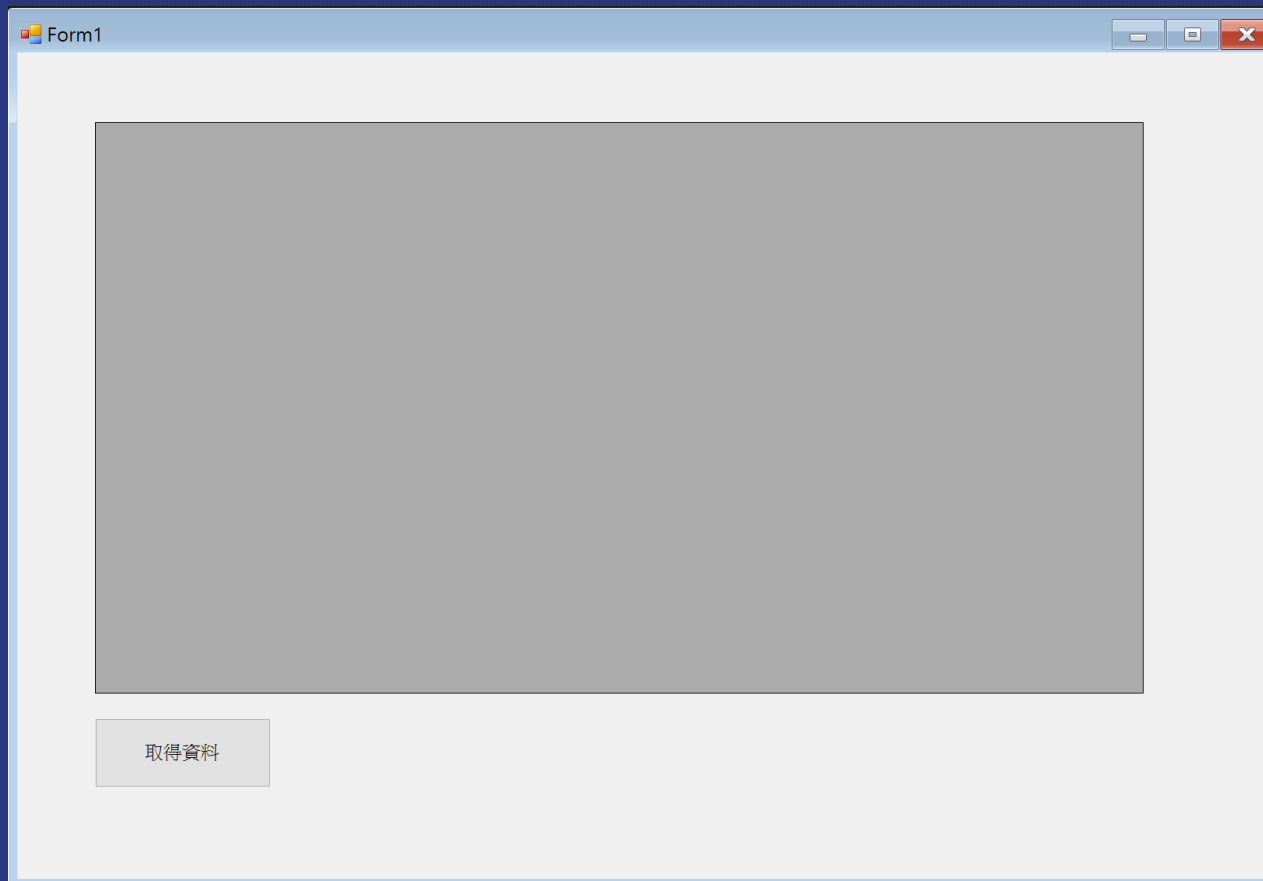
從網路取得資料的方式

- 命名空間：`System.Net.Http`
- 類別：`HttpClient`
 - `GetStringAsync` 方法
 - 這是一個特別的方法，被稱為非同步方法

LAB

使用 HttpClient

畫面配置



加入一個
DataGridView
一個 Button

DataGridView 的配置

- 請依據剛剛做的 Restaurant 類別配置 DataGridView
- AlsoKnowns 和 awards 不要配置

修改 Form1 Class ,
並且為 button1.Click 事件掛上以下的方法

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        dataGridView1.AutoGenerateColumns = false;
    }

    async private void button1_Click(object sender, EventArgs e)
    {
        HttpClient client = new HttpClient();
        // 這個就是剛剛在公開資料平台的資料下載網址
        string uri =
            "https://lod2.apc.gov.tw/API/v1/dump/datastore/A53000000A-000003-001";
        string data = await client.GetStringAsync(uri);
        var result = JsonConvert.DeserializeObject<Class1[]>(data);
        dataGridView1.DataSource = result[0].result.records ;
    }
}
```

執行

在這裏你將學到

Learn How to Learn

- 學新東西、新技術的能力
- 尋找解答的能力
- 隨時吸取新知識的能力

跟著你一輩子的能力 ...