跨平台 Hybrid App 開發簡介

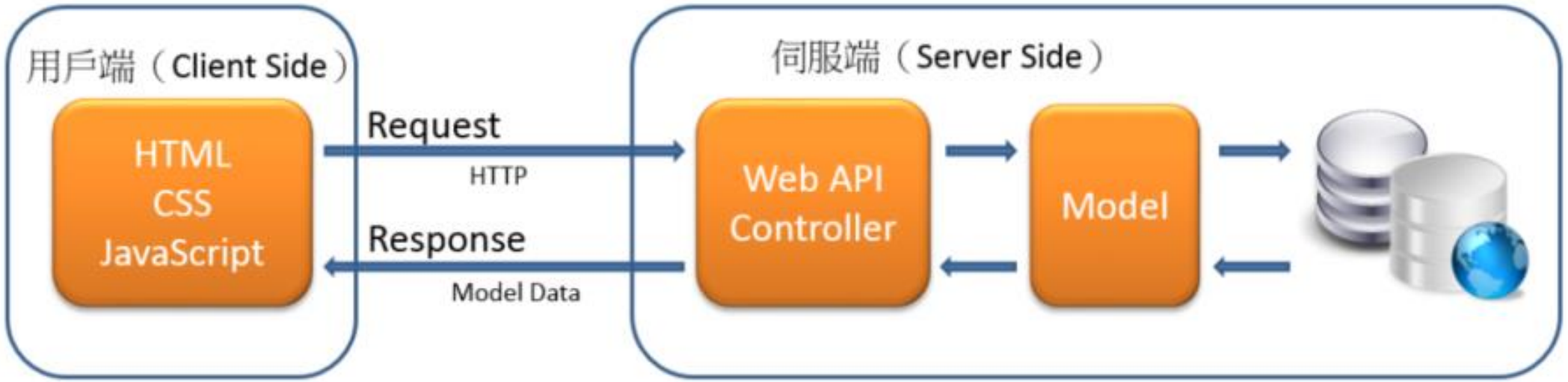
Jacky

前端所面對的挑戰

- 目前市面上常見的跨平台行動裝置的開發方案目前有三種主要工具:
  - Unity : 專業的遊戲開發引擎最接近原生ＡＰＰ
  - Xamarin : C# 語言為主的 Mono專案開發套件
  - Cordova : HTML/CSS/Javascript來開發跨平台APP語言為主的 Cordova /PhoneGap
- 基於已經有網頁開發的背景， 選用 HTML5 也就是混合式行動裝置應用程式（ Hybrid Mobile Apps ） 或許是最好且最快的選擇。

	Mobile_web	Hybrid	croaa_compiling	Native_app
代表性作品	html5/jquery	phoneGap	Xamarin	objective_c/java
跨平台能力	強	強	中	低
使用者經驗	中	中	強	強
離線能力	無	有	有	有
整合性與功能	弱	有	高	高
學習門檻	低	低	中	高
效能	低	中->高	中->高	高

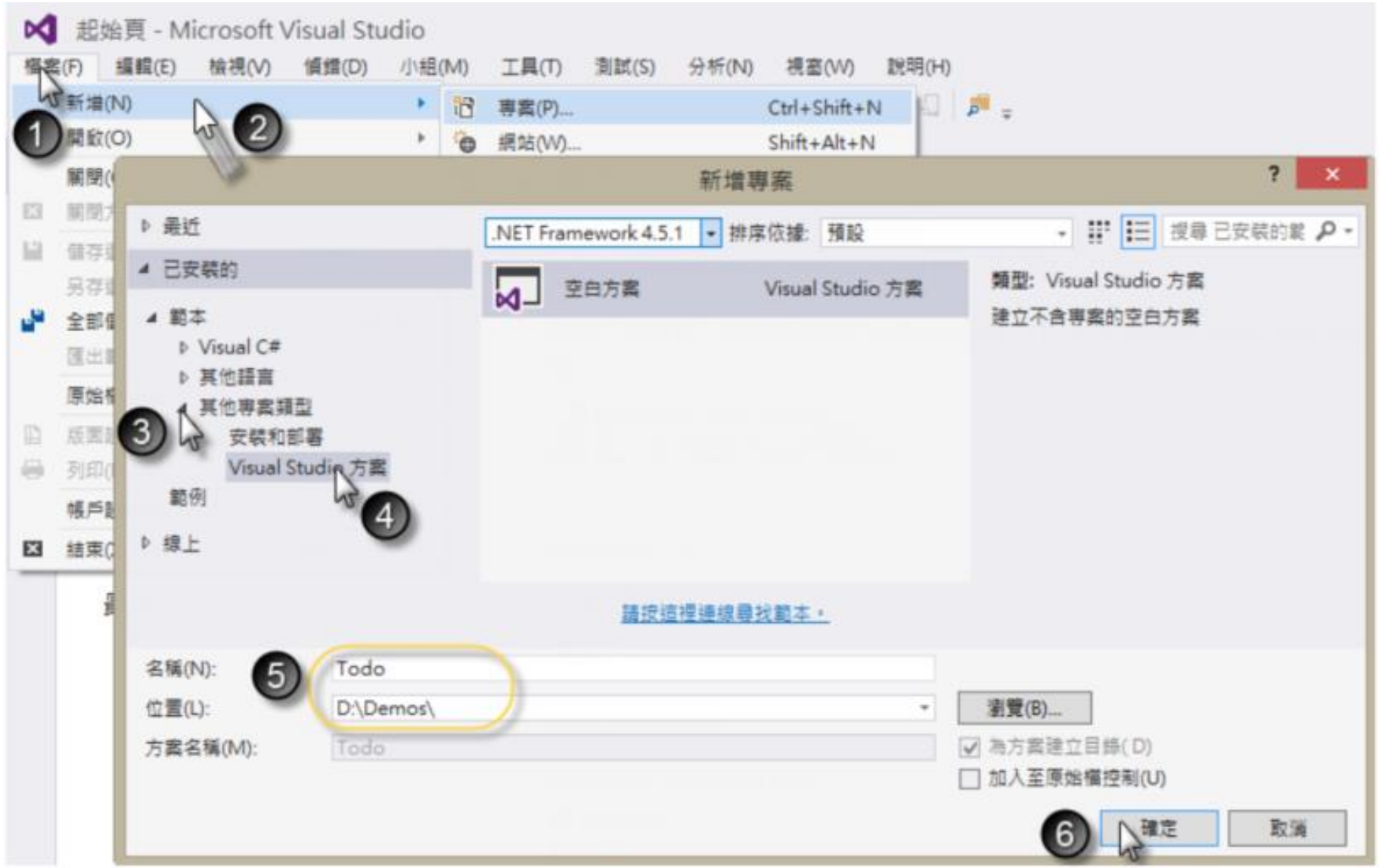
後端技術的選擇


用戶端（Client Side）
HTML
CSS
JavaScript
Request
HTTP
Response
Model Data
伺服端（Server Side）
Web API
Controller
Model

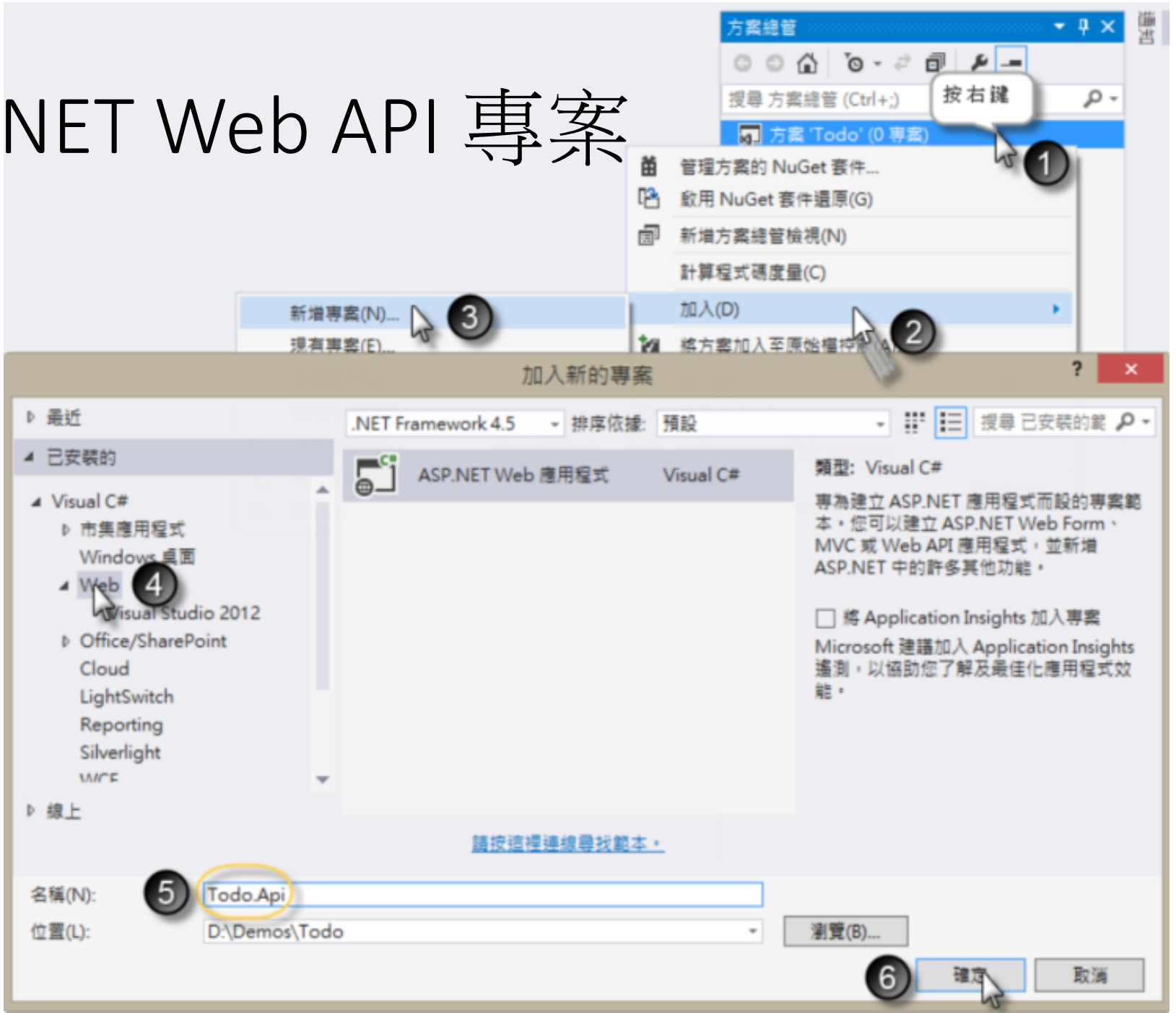
需求案例:行動版待辦事項

- 老公是個健忘的人， 老婆交待要買的東西不是經常忘記就是買錯了， 而老婆是個性急的人， 一想到要買什麼東西， 馬上就會打電話交待老公下班時買回來。
- 這看來是一個大問題， 讓我們一起來幫他們解決難題吧！

建立空白方案

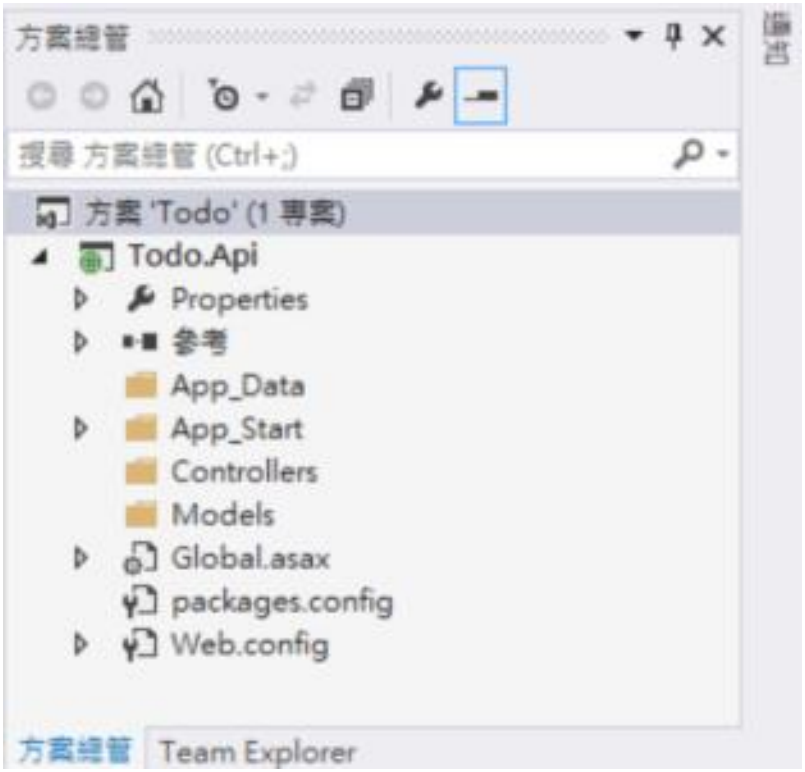

起始頁 - Microsoft Visual Studio
檔案(F) 編輯(E) 檢視(V) 偵錯(D) 小組(M) 工具(T) 測試(S) 分析(N) 視窗(W) 說明(H)
新增(N)
開啟(O)
專案(P)... Ctrl+Shift+N
網站(W)... Shift+Alt+N
1
2
新增專案
最近
已安裝的
範本
Visual C#
其他語言
其他專案類型
安裝和部署
Visual Studio 方案
範例
線上
3
4
.NET Framework 4.5.1
排序依據: 預設
空白方案 Visual Studio 方案
類型: Visual Studio 方案
建立不含專案的空白方案
請按這裡連線尋找範本。
名稱(N): Todo
位置(L): D:\Demos\
方案名稱(M): Todo
5
瀏覽(B)...
為方案建立目錄(D)
加入至原始檔控制(U)
6
確定
取消

建立 ASP.NET Web API 專案

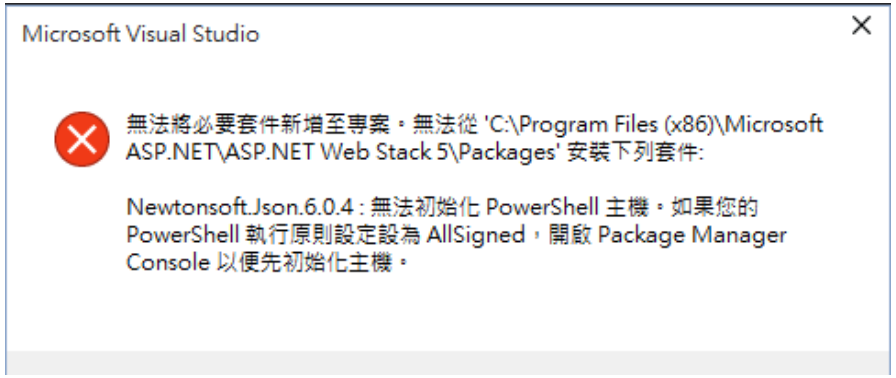


專案設定：空白專案


新增 ASP.NET 專案 - Todo.Api
選取範本(S):
Empty
Web Forms
MVC
Web API
Single Page Application
Azure Mobile Service
建立 ASP.NET 應用程式的空專案範本，此範本沒有任何內容。
深入了解
變更驗證(A)
驗證: 不驗證
Microsoft Azure
雲端中的主機(H)
網站
管理訂閱
為下列項目新增資料夾及核心參考:
Web Forms
MVC
Web API
新增單元測試
測試專案名稱:
Todo.Api.Tests
確定
取消
7
8
9


方案總管
搜尋 方案總管 (Ctrl+;)
方案 'Todo' (1 專案)
Todo.Api
Properties
參考
App_Data
App_Start
Controllers
Models
Global.asax
packages.config
Web.config
方案總管
Team Explorer

遇到怪問題---仍無正解...

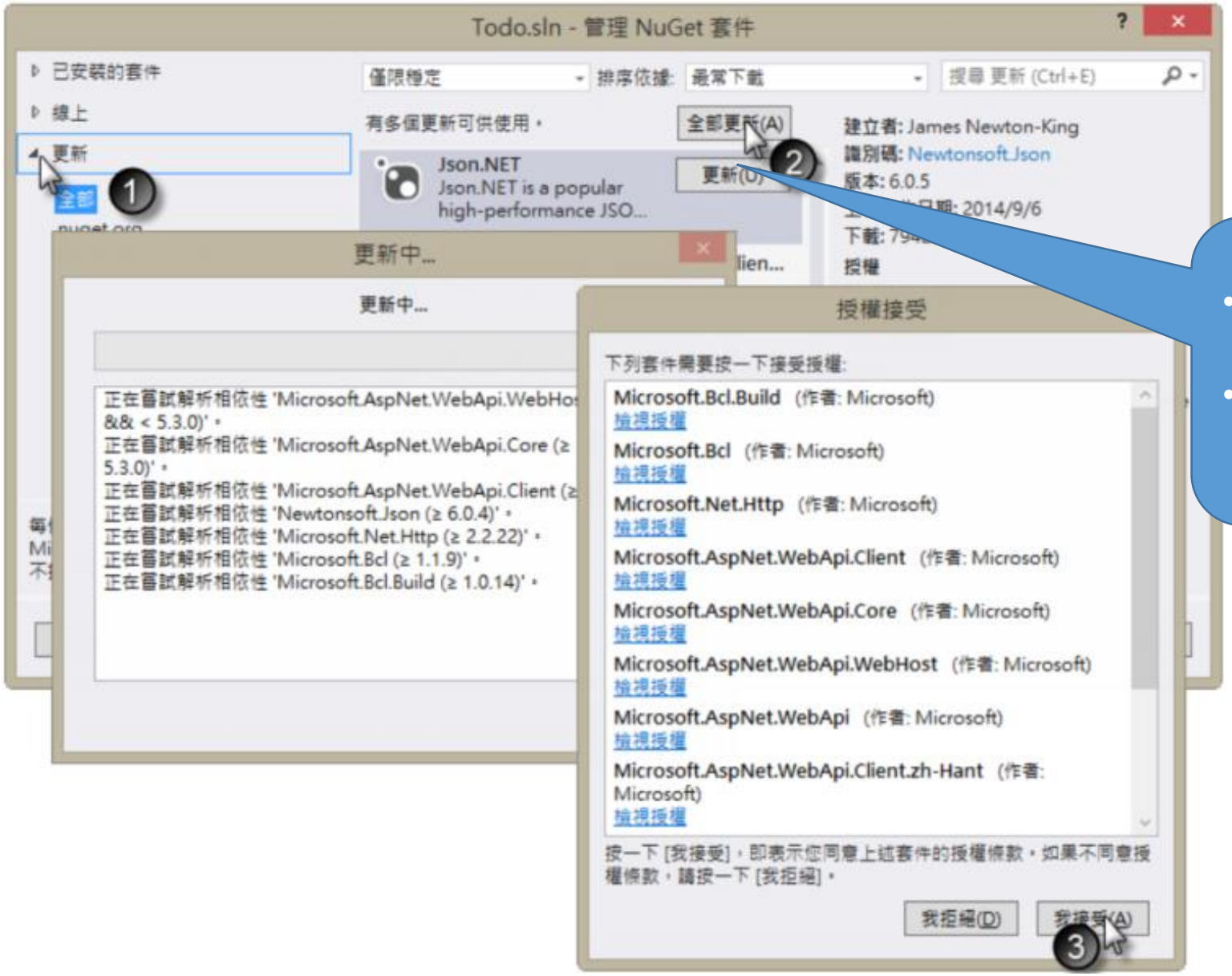


MayBe Try……

它是請我們將PowerShell execution policy設定成AllSigned，

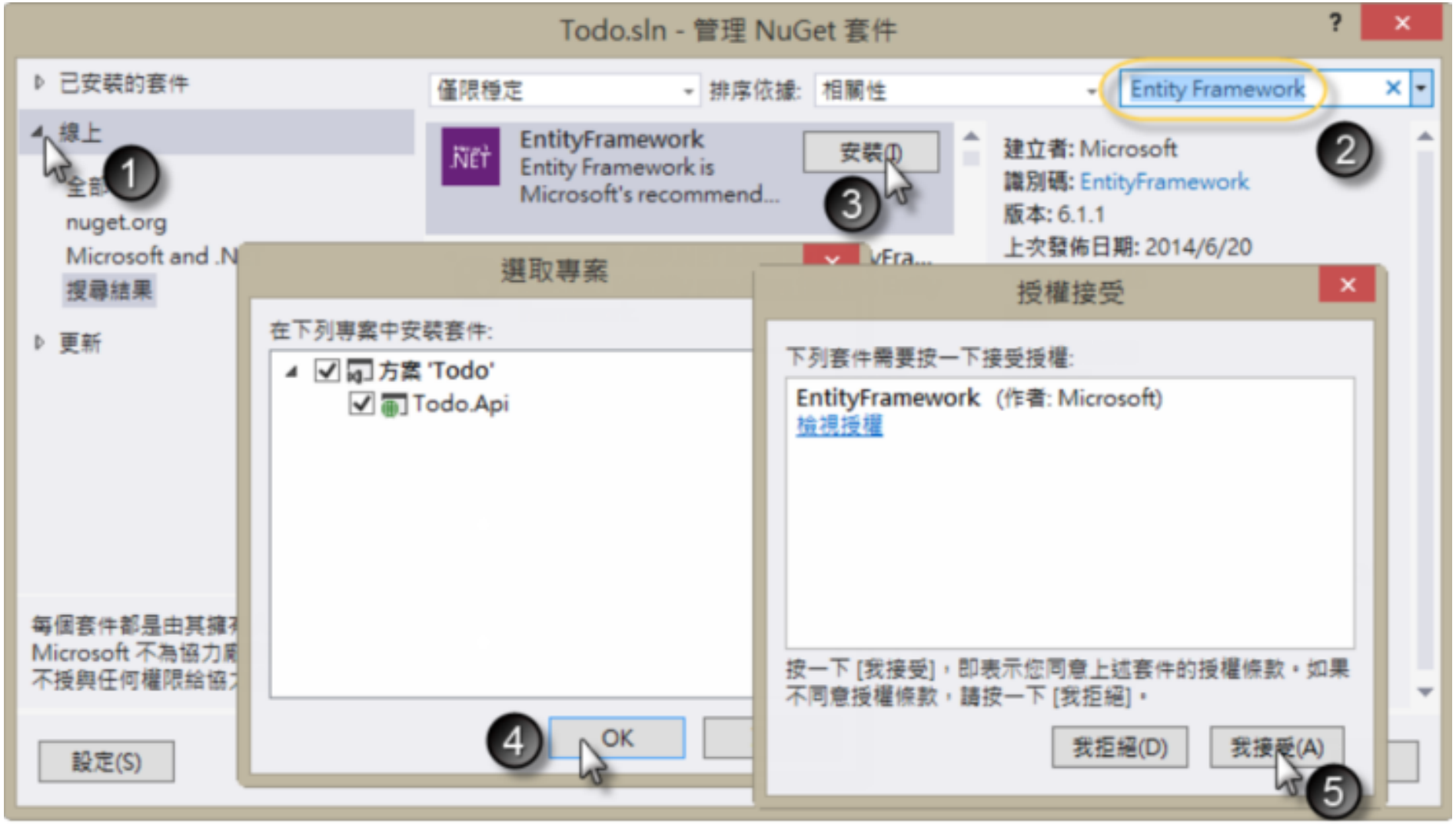
所以這時請以系統管理者身分開啟PowerShell，然後將PowerShell execution policy設定成AllSigned就可以了，如下，

NuGet 套件管理員-管理方案套件管理



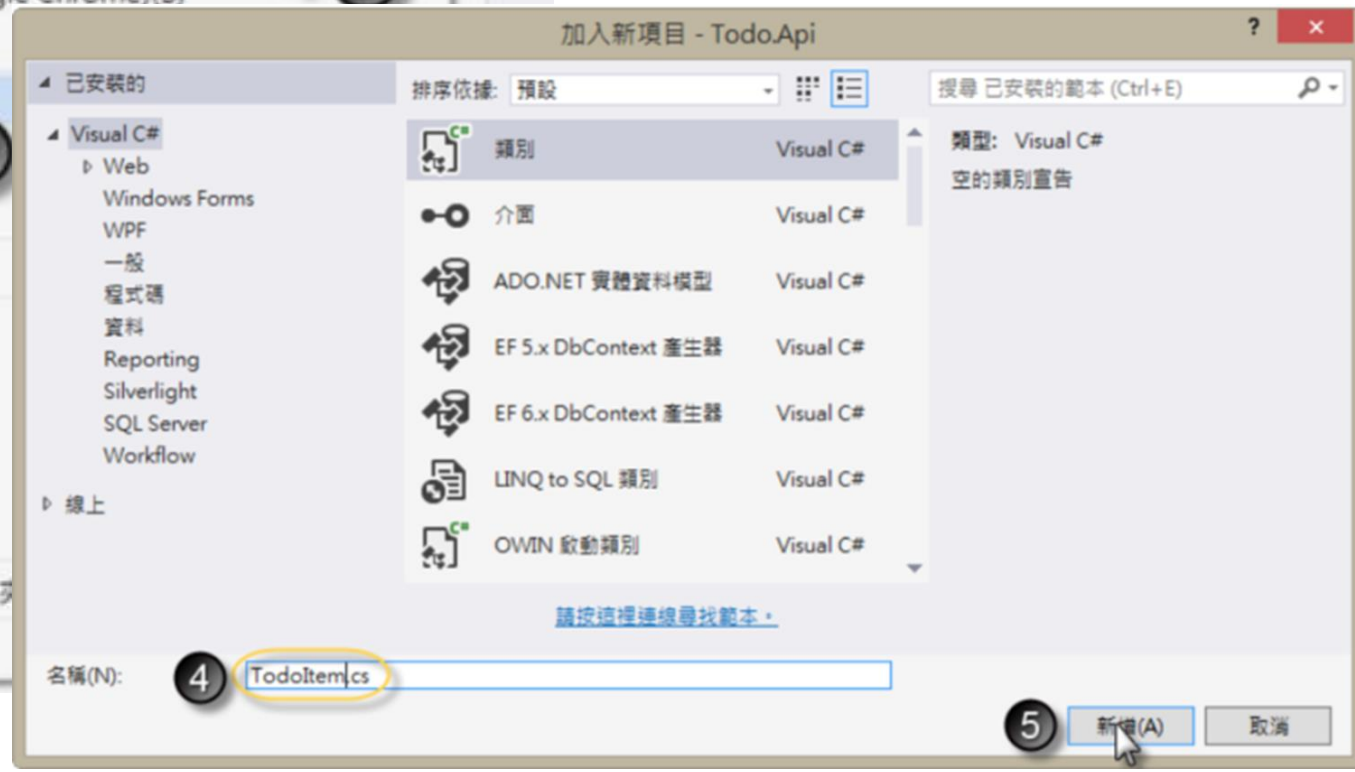
- [已安裝]→[全部]:可解除不需要的套件
- [更新]→[全部]:更新用到所有已安裝的套件

Entity Framework 套件安裝進來



加入資料模型

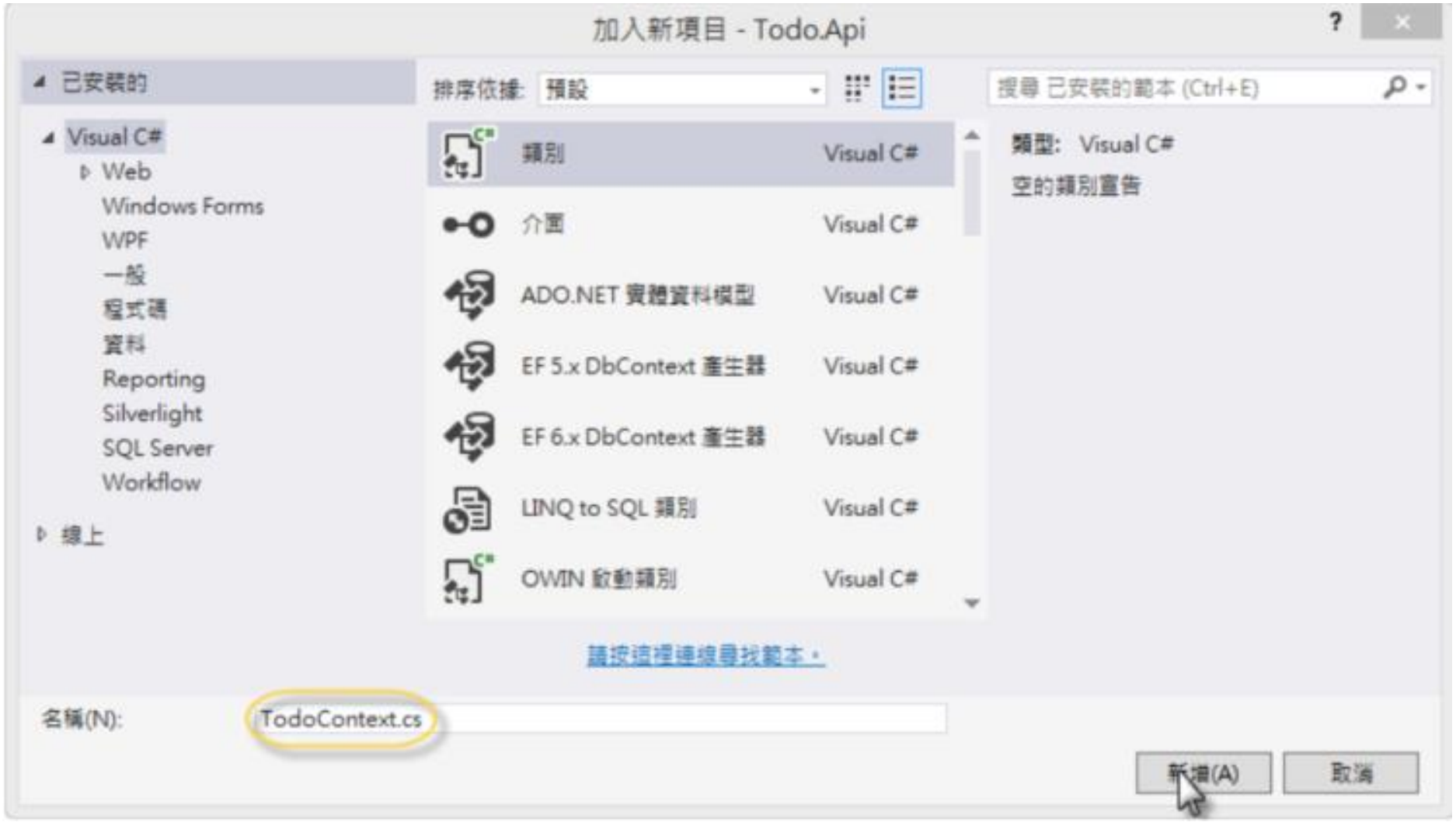

方案總管
搜尋 方案總管 (Ctrl+;)
方案 'Todo' (1 專案)
Todo.Api
Properties
參考
App_Data
Models
按右鍵
1
在瀏覽器中檢視 (Google Chrome)(B)
瀏覽方式(H)...
加入(D)
2
新增方案總管檢視(N)
從專案移除(J)
剪下(T)
複製(Y)
貼上(P)
刪除(D)
重新命名(M)
在檔案總管中開啟資料夾
屬性(R)
控制器...
新增項目(W)... Ctrl+Shift+A
現有項目(G)... Shift+Alt+A
新增 Scaffolded 項目...
新增資料夾(D)
加入 ASP.NET 資料夾(S)
新增 Azure WebJob 專案
將現有的專案做為 Azure WebJob
類別(C)... Shift+Alt+C
3


加入新項目 - Todo.Api
已安裝的
Visual C#
Web
Windows Forms
WPF
一般
程式碼
資料
Reporting
Silverlight
SQL Server
Workflow
線上
排序依據: 預設
類別 Visual C#
介面 Visual C#
ADO.NET 實體資料模型 Visual C#
EF 5.x DbContext 產生器 Visual C#
EF 6.x DbContext 產生器 Visual C#
LINQ to SQL 類別 Visual C#
OWIN 啟動類別 Visual C#
搜尋 已安裝的範本 (Ctrl+E)
類型: Visual C#
空的類別宣告
請按這裡連線尋找範本。
名稱(N): TodoItem.cs
4
5
新增(A)
取消

資料模型加入程式碼

```
using System. ComponentModel. DataAnnotations;

namespace Todo. Api. Models
{
public class TodoItem
    {
        public int Id { get; set; }
        [ Required, StringLength( maximumLength: 30) ]
        public string Description { get; set; }
        public bool IsDone { get; set; }
    }
}
```

DbContext


加入新項目 - Todo.Api
已安裝的
Visual C#
Web
Windows Forms
WPF
一般
程式碼
資料
Reporting
Silverlight
SQL Server
Workflow
線上
排序依據: 預設
搜尋 已安裝的範本 (Ctrl+E)
類別 Visual C#
介面 Visual C#
ADO.NET 實體資料模型 Visual C#
EF 5.x DbContext 產生器 Visual C#
EF 6.x DbContext 產生器 Visual C#
LINQ to SQL 類別 Visual C#
OWIN 啟動類別 Visual C#
類型: Visual C#
空的類別宣告
請按這裡連線尋找範本。
名稱(N):
TodoContext.cs
新增(A)
取消

DbContext加入程式碼

```
using System. Data. Entity;

namespace Todo. Api. Models
{
        public class TodoContext : DbContext
        {
                public DbSet< TodoItem> TodoItems { get; set; }
        }
}
```

資料庫

Web.config

…….

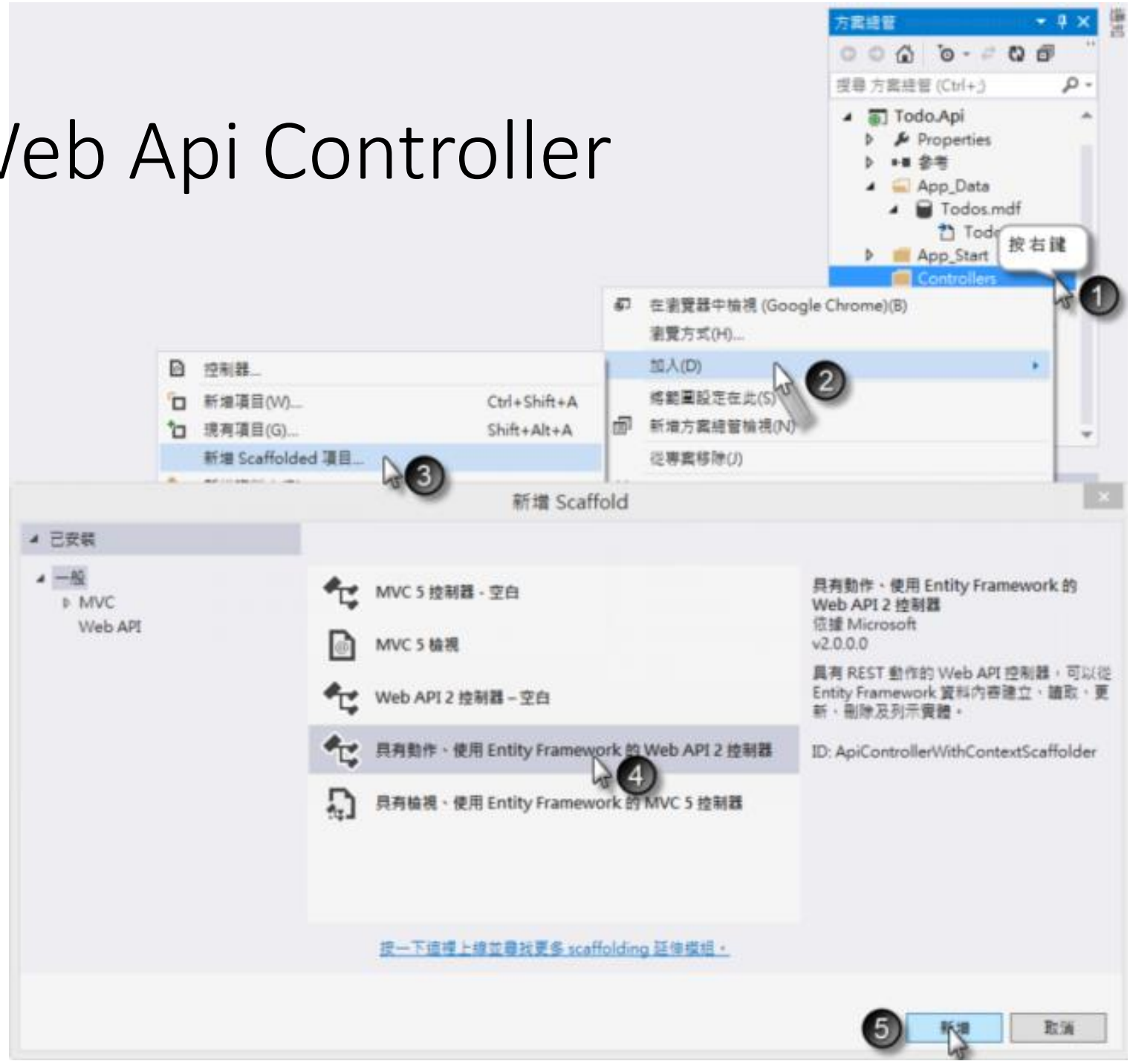
</configSections>

```
 <connectionStrings>
      <add name="TodoContext"
          providerName="System.Data.SqlClient"
          connectionString="Data Source=(LocalDB)\v11.0;Initial Catalog=Todos;
          Integrated Security=True;MultipleActiveResultSets=True" />
   </connectionStrings>
```

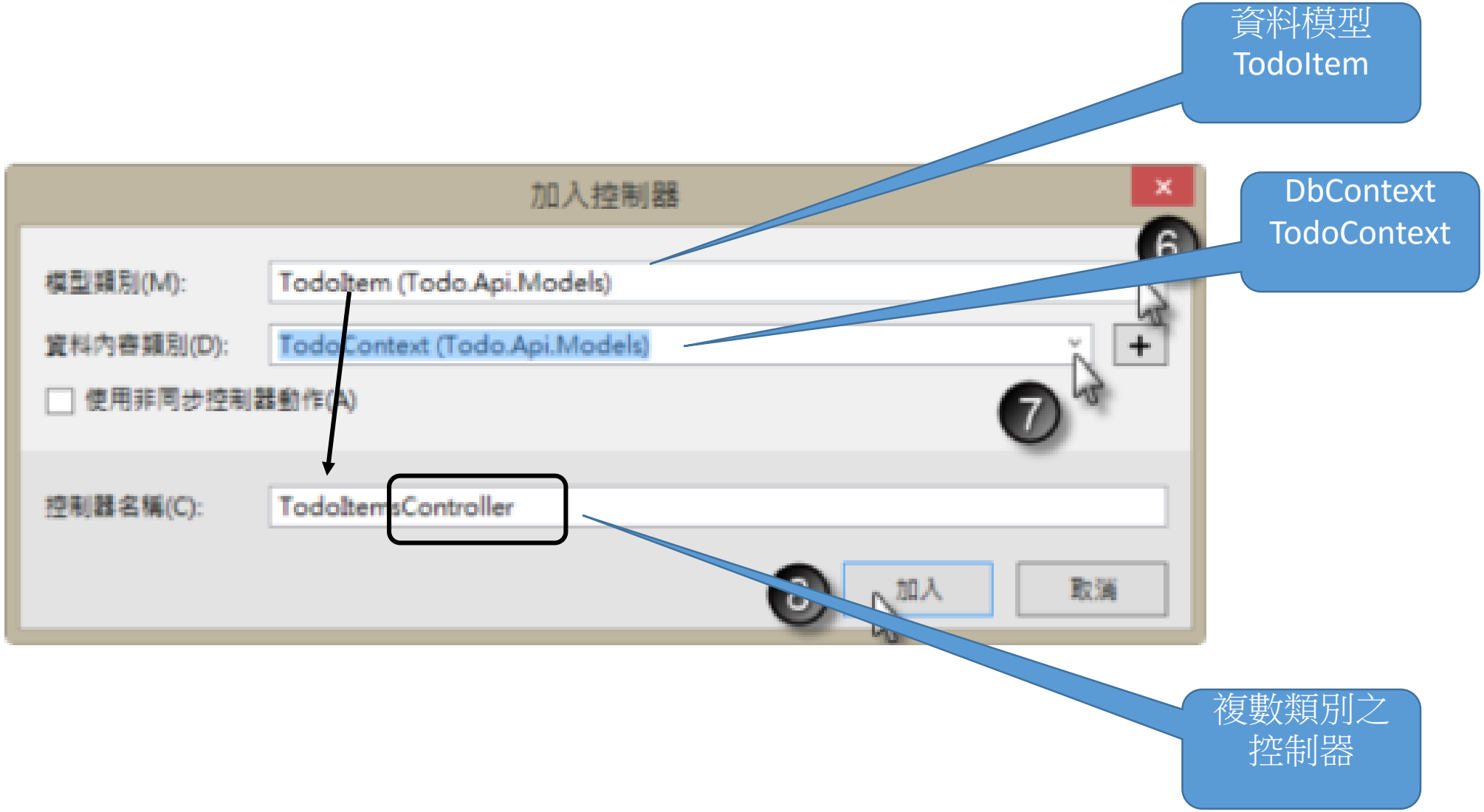
<appSettings></appSettings>

…….

Web Api Controller

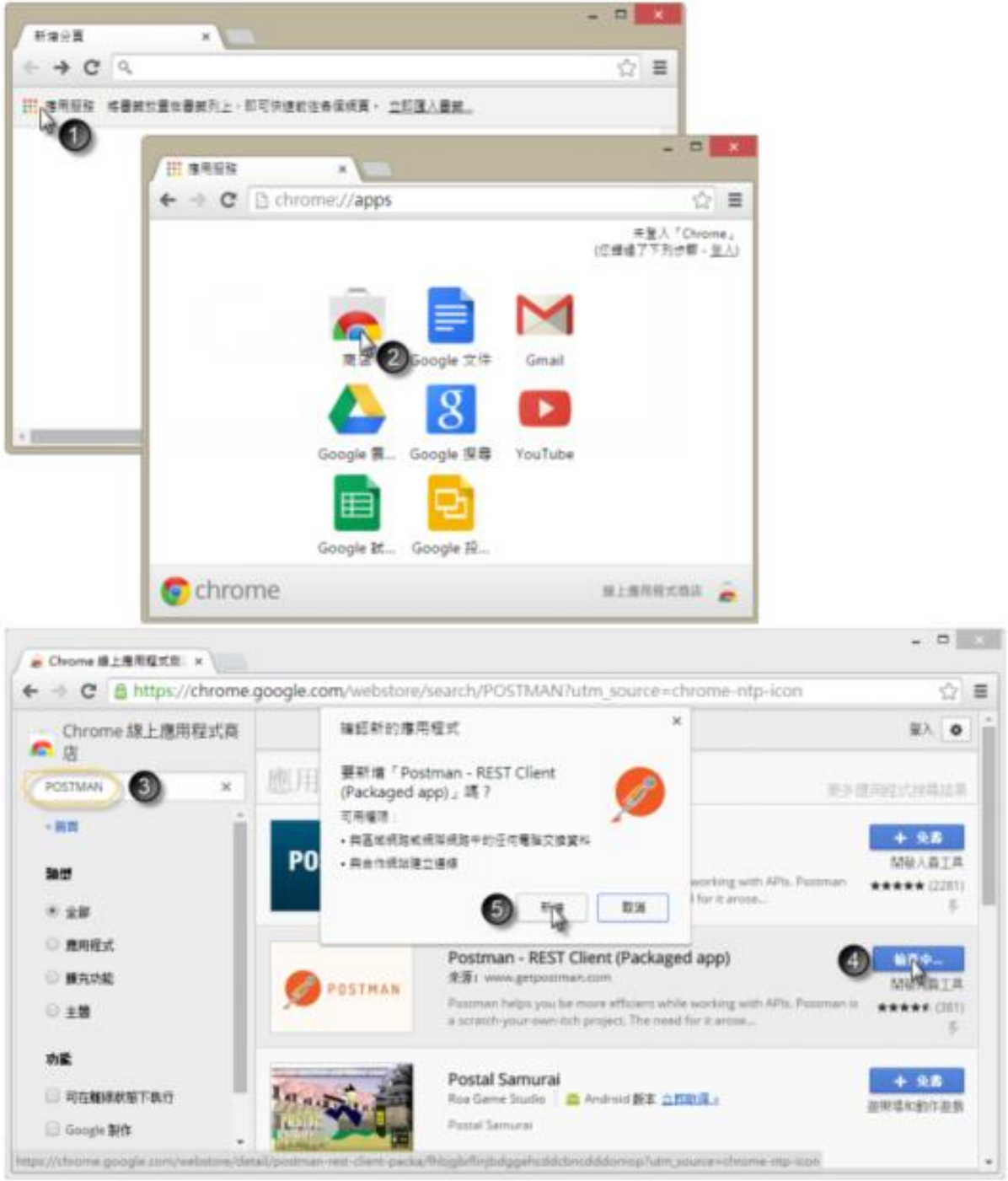


加入控制器


資料模型
TodoItem
DbContext
TodoContext
加入控制器
模型類別(M):
TodoItem (Todo.Api.Models)
資料內容類別(D):
TodoContext (Todo.Api.Models)
使用非同步控制器動作(A)
控制器名稱(C):
TodoItemsController
加入
取消
6
7
8
複數類別之
控制器

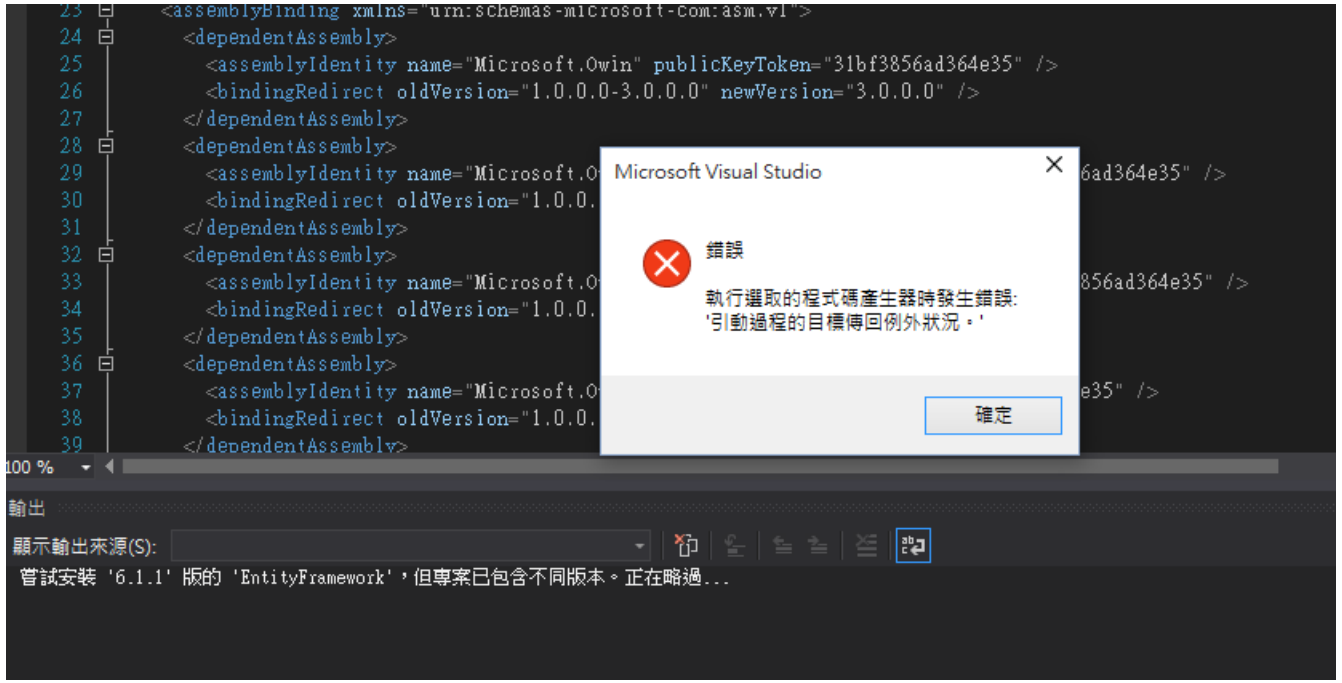
測試 Web API

- 安裝Google Chrome 擴充功能的 POSTMAN 工具

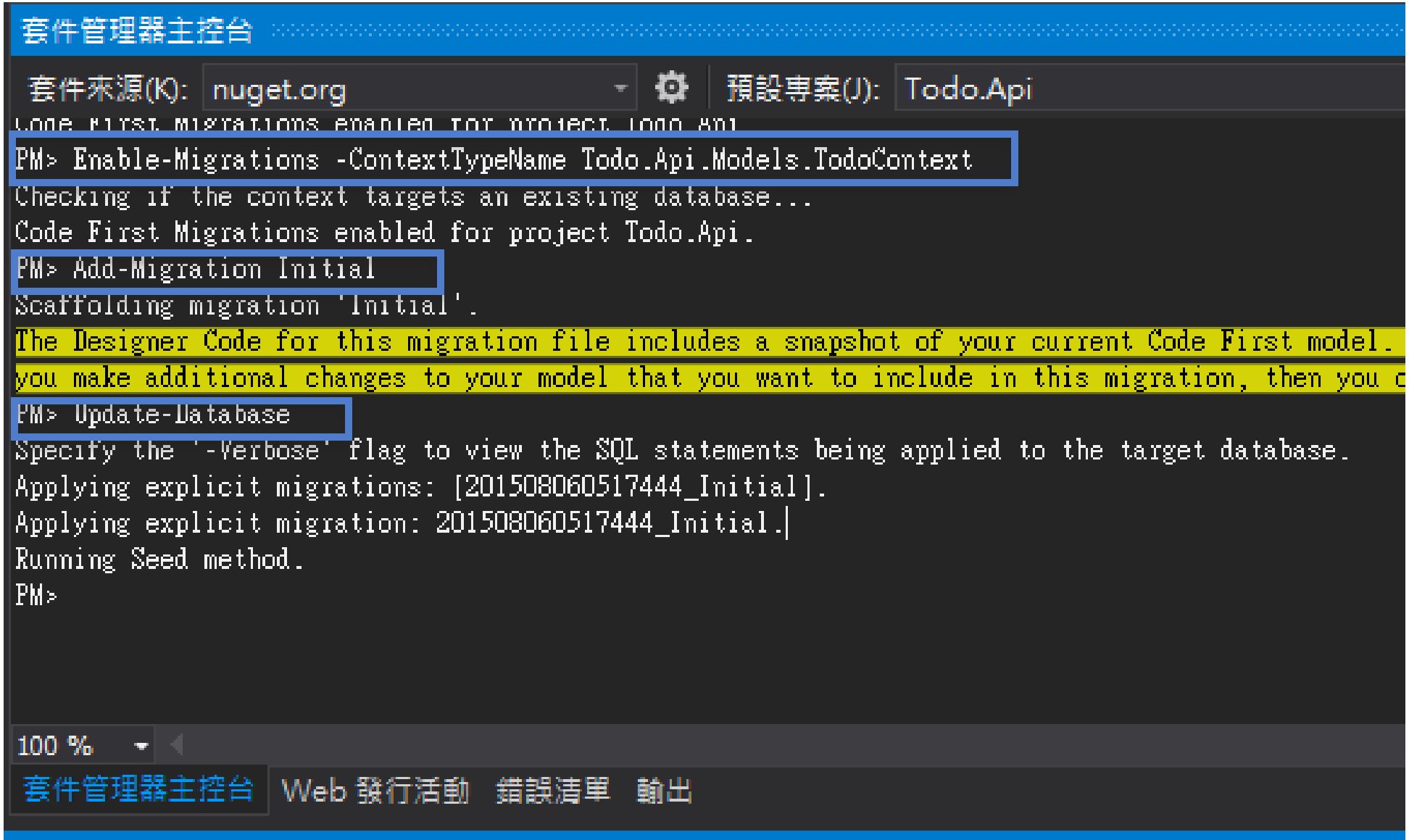


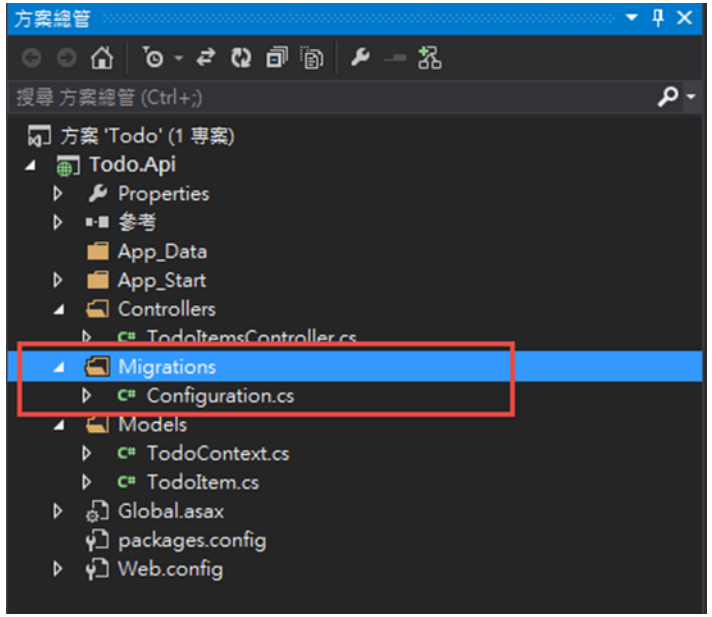
又一怪問題

檢查Web.config 資料庫連接字串



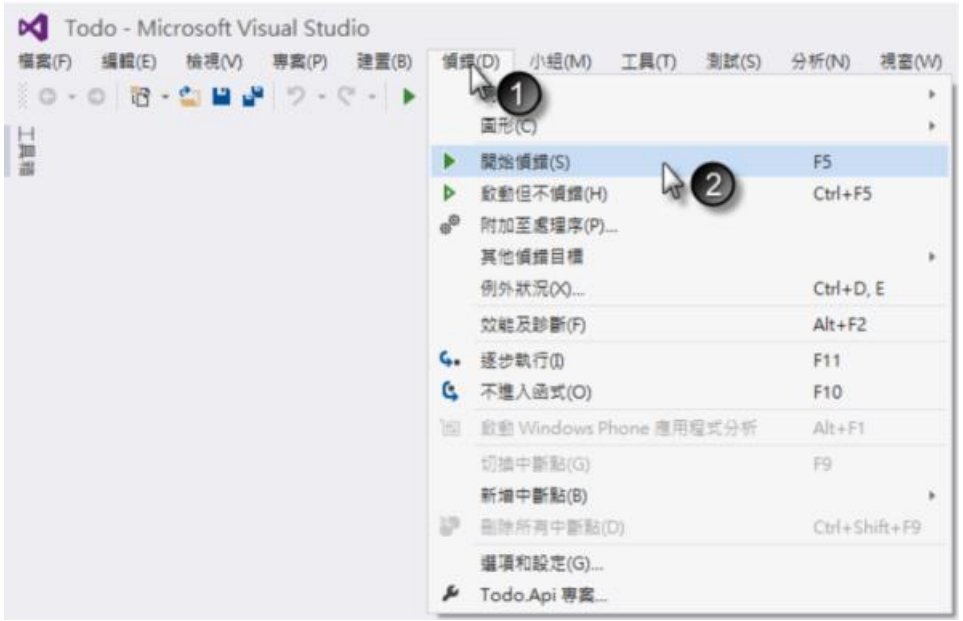
啟用資料移轉功能

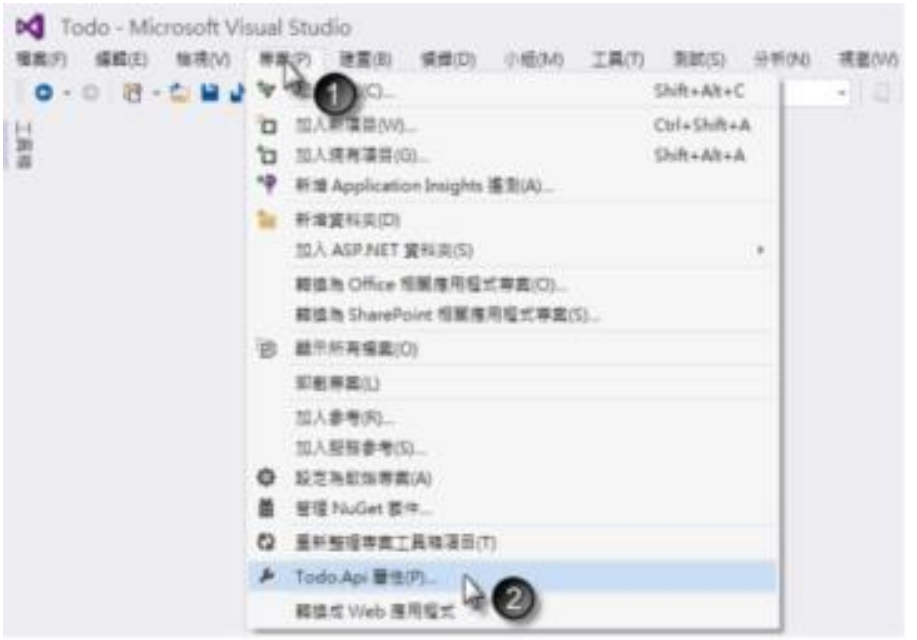

套件管理器主控台
套件來源(K): nuget.org
預設專案(J): Todo.Api
PM> Enable-Migrations -ContextTypeName Todo.Api.Models.TodoContext
Checking if the context targets an existing database...
Code First Migrations enabled for project Todo.Api.
PM> Add-Migration Initial
Scaffolding migration 'Initial'.
The Designer Code for this migration file includes a snapshot of your current Code First model.
you make additional changes to your model that you want to include in this migration, then you c
PM> Update-Database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
Applying explicit migrations: [201508060517444_Initial].
Applying explicit migration: 201508060517444_Initial.
Running Seed method.
PM>
100 %
套件管理器主控台
Web 發行活動
錯誤清單
輸出



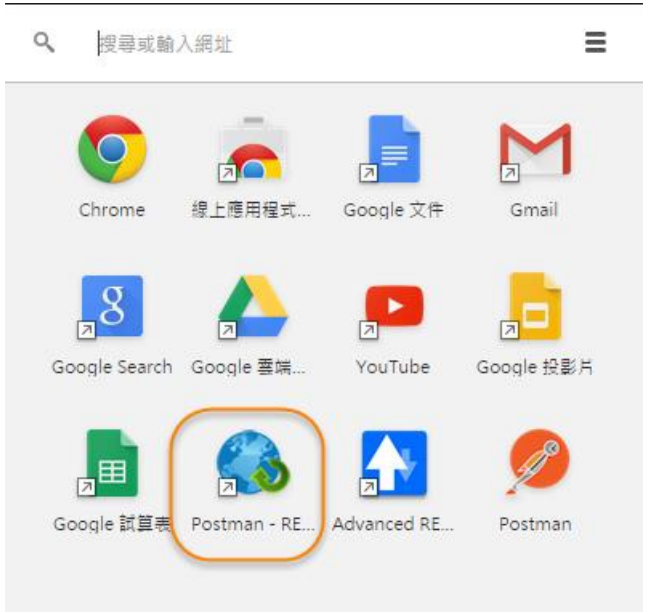

```
namespace Todo.Api.Migrations
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Migrations;
    using System.Linq;

    internal sealed class Configuration : DbMigrationsConfiguration<Todo.Api.Models.TodoContext>
    {
        public Configuration()
        {
            AutomaticMigrationsEnabled = false;
        }

        protected override void Seed(Todo.Api.Models.TodoContext context)
        {
            //  This method will be called after migrating to the latest version.

            //  You can use the DbSet<T>.AddOrUpdate() helper extension method
            //  to avoid creating duplicate seed data. E.g.
            //
            //    context.People.AddOrUpdate(
            //      p => p.FullName,
            //      new Person { FullName = "Andrew Peters" },
            //      new Person { FullName = "Brice Lambson" },
            //      new Person { FullName = "Rowan Miller" }
            //    );
            //
        }
    }
```

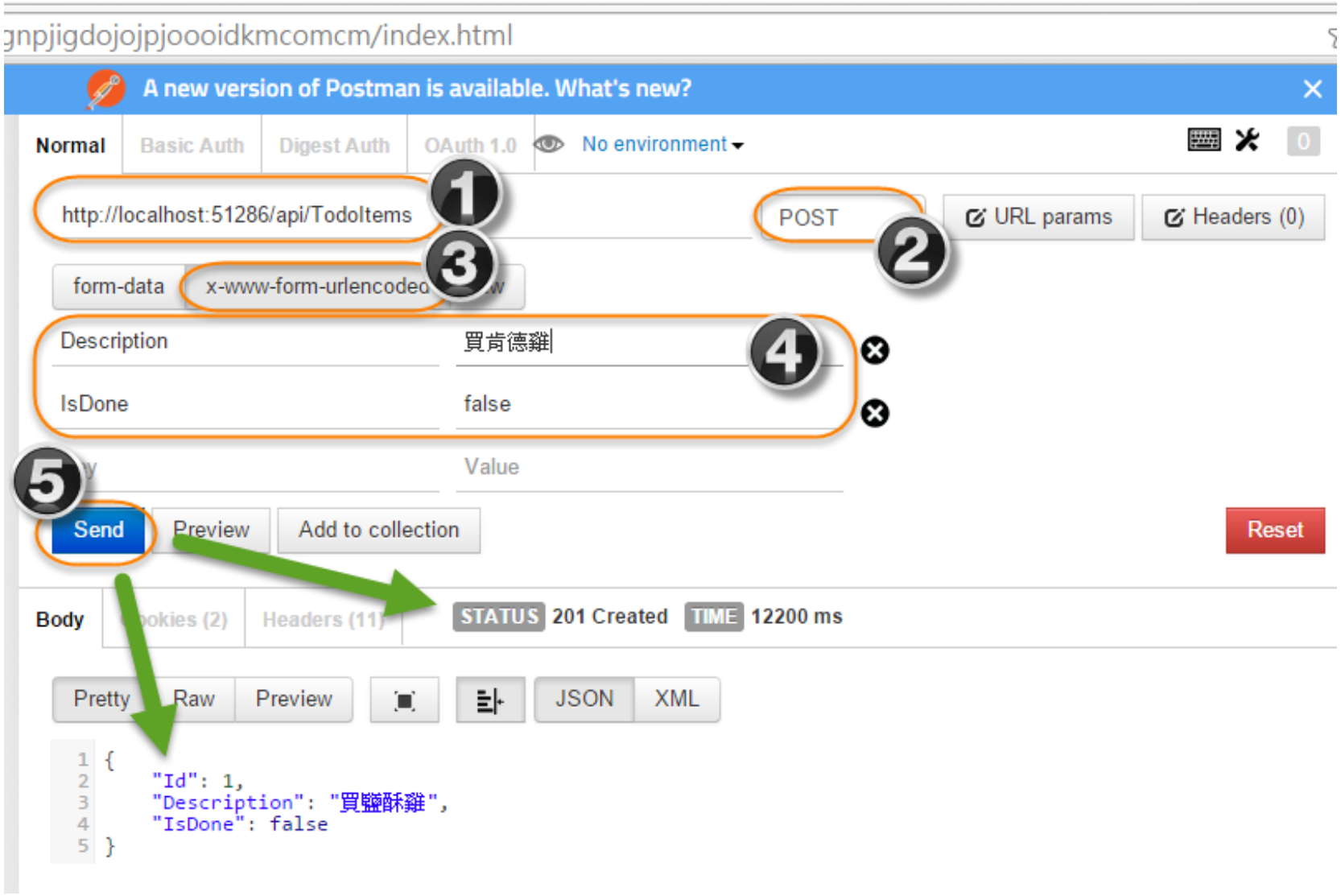
取得目前WebAPI專案的 URL





啟動 Postman 測試工具

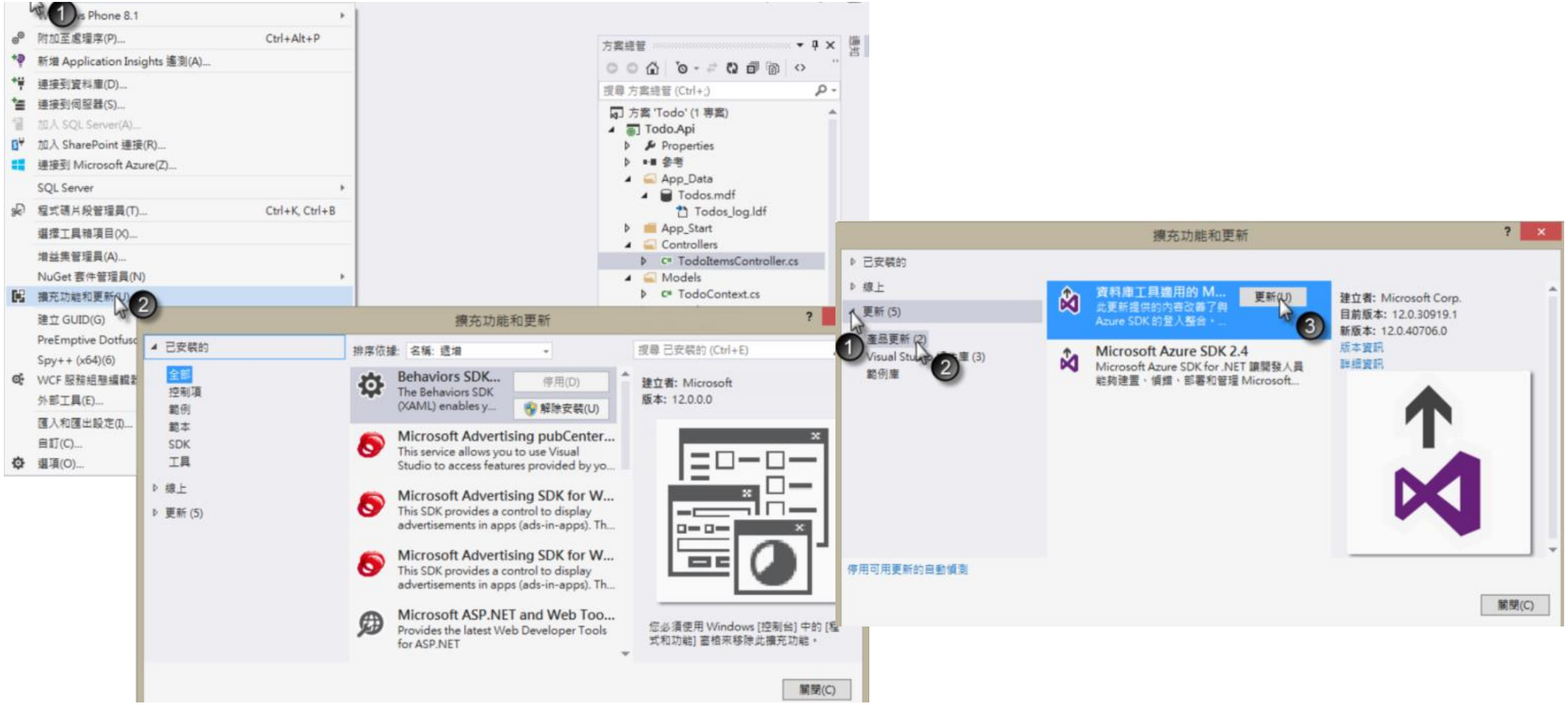




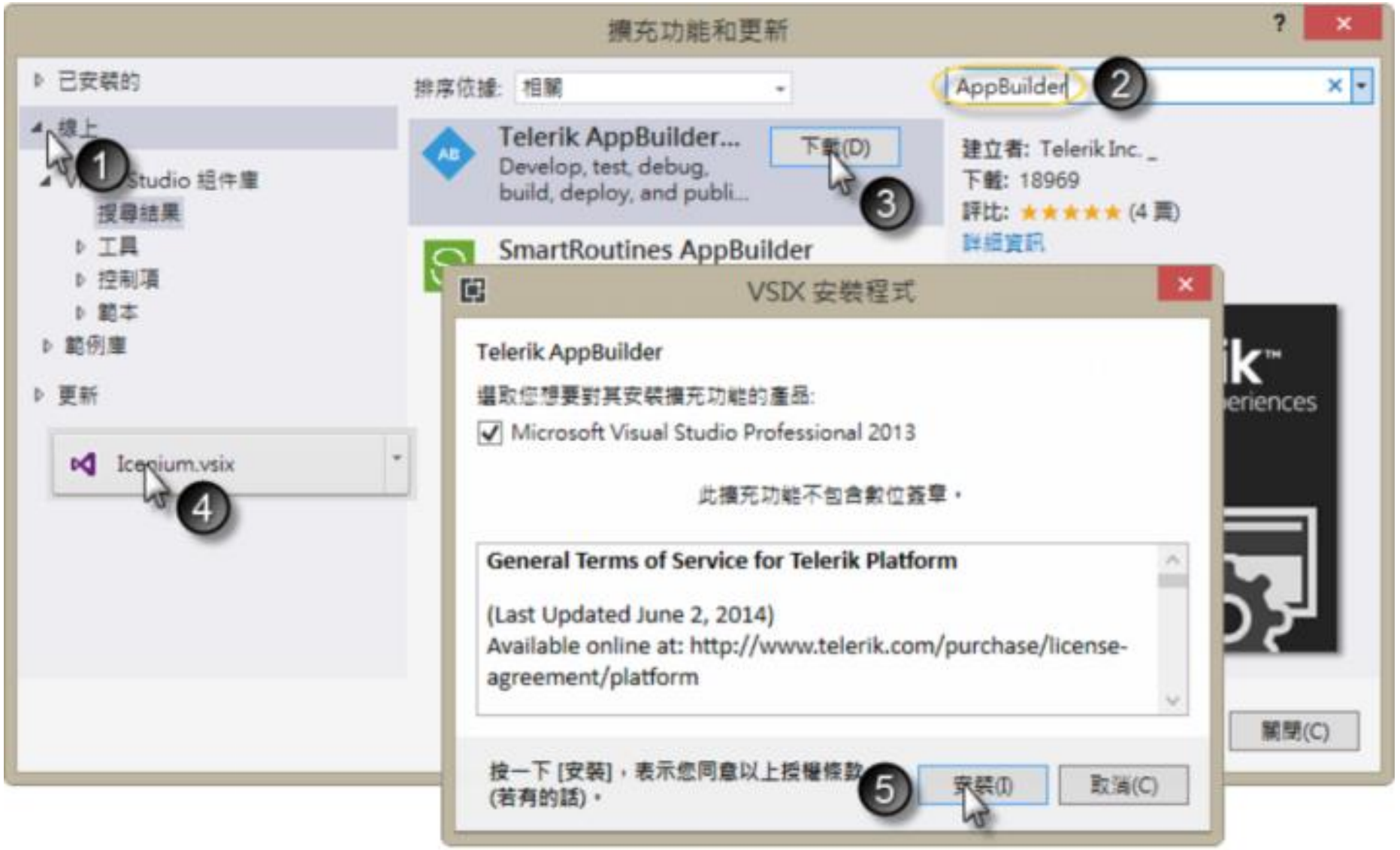
依序測試C新增R查詢U修改D刪除

C	POST	http://localhost:51286/api/TodoItems
R	GET	http://localhost:51286/api/TodoItems
U	PUT	http://localhost:51286/api/TodoItems/1
D	DELETE	http://localhost:51286/api/TodoItems/1

擴充功能和更新

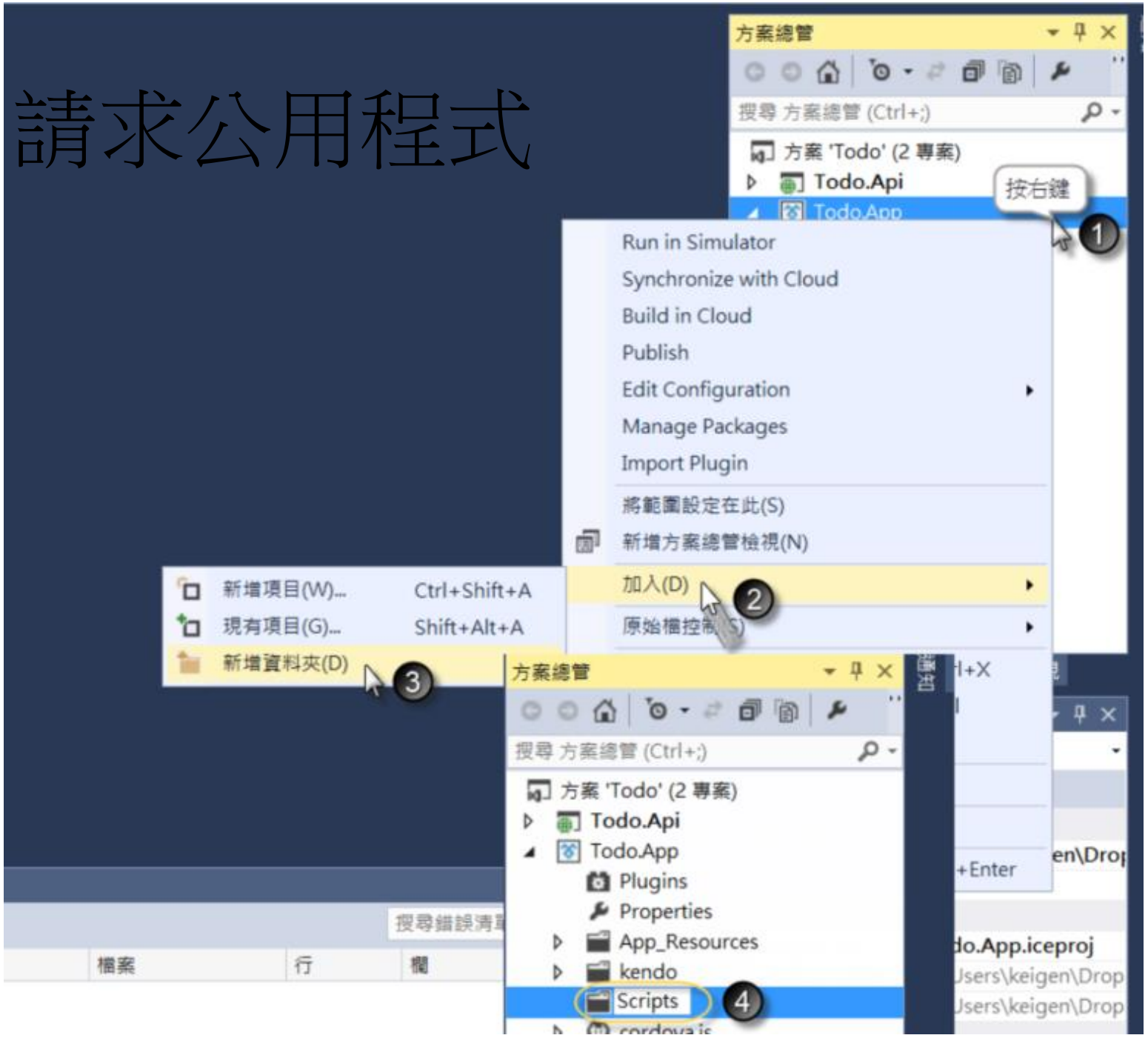

Phone 8.1
附加至處理序(P)... Ctrl+Alt+P
新增 Application Insights 遙測(A)...
連接到資料庫(D)...
連接到伺服器(S)...
加入 SQL Server(A)...
加入 SharePoint 連接(R)...
連接到 Microsoft Azure(Z)...
SQL Server
程式碼片段管理員(T)... Ctrl+K, Ctrl+B
選擇工具箱項目(X)...
增益集管理員(A)...
NuGet 套件管理員(N)
擴充功能和更新(U)...
建立 GUID(G)
PreEmptive Dotfuscator
Spy++ (x64)(6)
WCF 服務組態編輯器
外部工具(E)...
匯入和匯出設定(I)...
自訂(C)...
選項(O)...
方案總管
搜尋 方案總管 (Ctrl+;)
方案 'Todo' (1 專案)
Todo.Api
Properties
參考
App_Data
Todos.mdf
Todos_log.ldf
App_Start
Controllers
TodoItemsController.cs
Models
TodoContext.cs
擴充功能和更新
已安裝的
全部
控制項
範例
範本
SDK
工具
線上
更新 (5)
排序依據: 名稱: 遞增
搜尋 已安裝的 (Ctrl+E)
Behaviors SDK...
The Behaviors SDK (XAML) enables y...
停用(D)
解除安裝(U)
Microsoft Advertising pubCenter...
This service allows you to use Visual Studio to access features provided by yo...
Microsoft Advertising SDK for W...
This SDK provides a control to display advertisements in apps (ads-in-apps). Th...
Microsoft Advertising SDK for W...
This SDK provides a control to display advertisements in apps (ads-in-apps). Th...
Microsoft ASP.NET and Web Too...
Provides the latest Web Developer Tools for ASP.NET
建立者: Microsoft
版本: 12.0.0.0
您必須使用 Windows [控制台] 中的 [程式和功能] 資格來移除此擴充功能。
關閉(C)
擴充功能和更新
已安裝的
線上
更新 (5)
產品更新 (2)
Visual Studio 擴充功能 (3)
範例庫
資料庫工具適用的 M...
此更新提供的內容改善了與 Azure SDK 的登入整合。...
更新(U)
Microsoft Azure SDK 2.4
Microsoft Azure SDK for .NET 讓開發人員能夠建置、偵錯、部署和管理 Microsoft...
建立者: Microsoft Corp.
目前版本: 12.0.30919.1
新版本: 12.0.40706.0
版本資訊
詳細資訊
停用可用更新的自動偵測
關閉(C)
1
2
3

加入AppBuilder擴充功能



建立 AppBuilder 專案

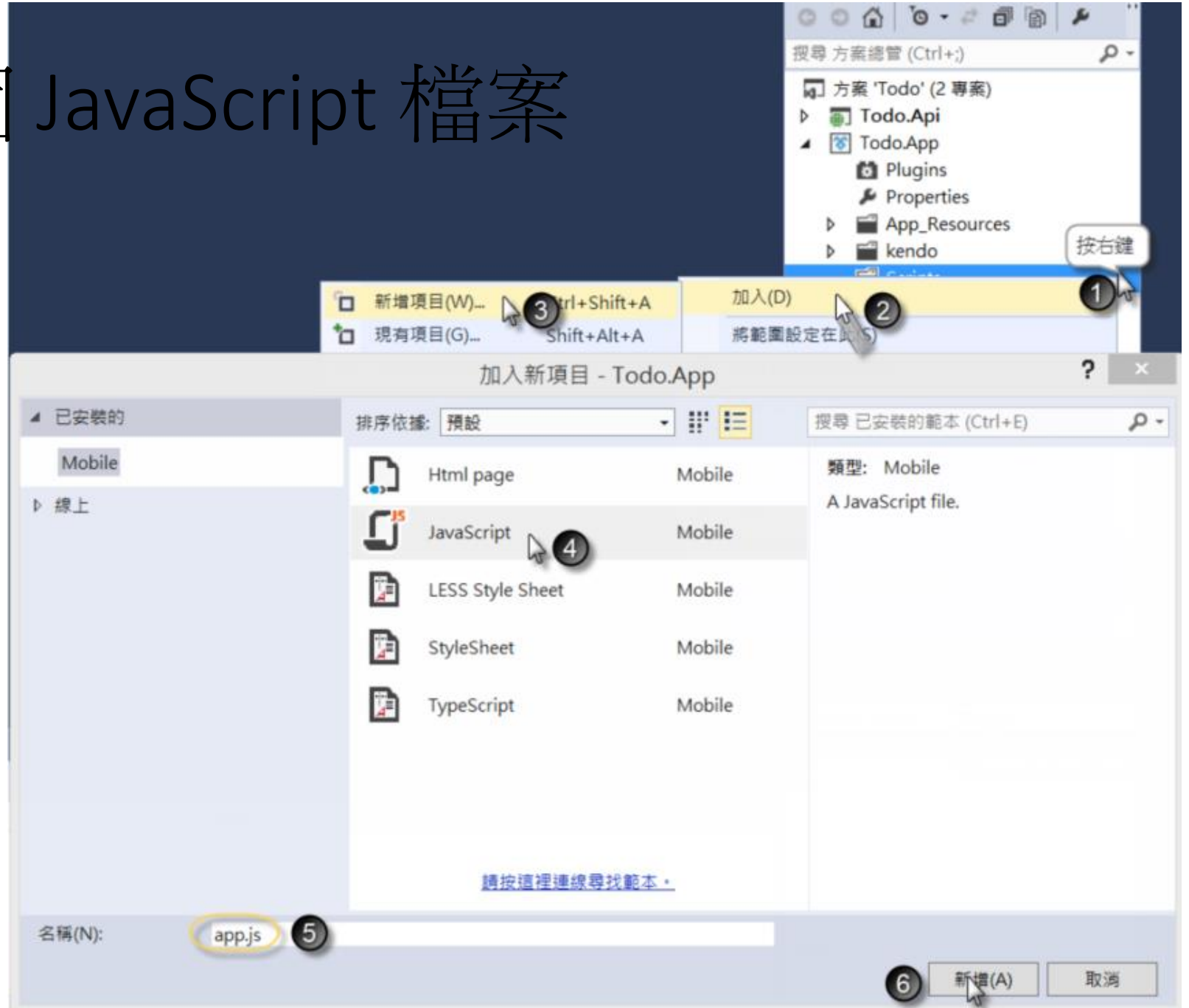
實作 Ajax 請求公用程式



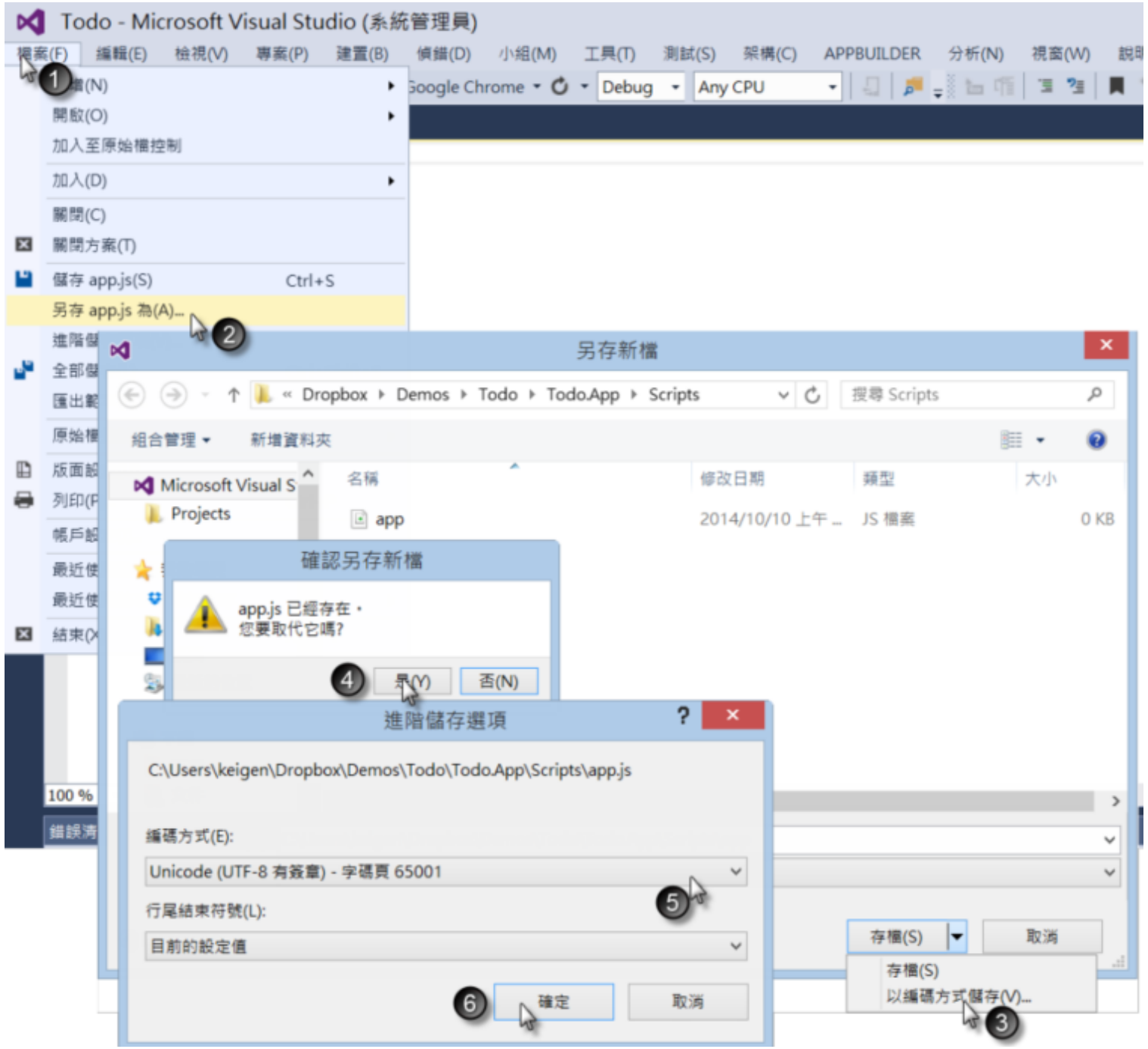
新增config JavaScript 檔案

```
(function (global) {
    var app = global.app = global.app || {};
     //發佈雲端
    //var serviceUrl = "http://jackytodo.azurewebsites.net/api/";
    //本機測試
    var serviceUrl = "http://localhost:50633/api/";

    app.config = {
        todoItemsUrl: serviceUrl + "TodoItems/"
    };
})(window)
```

新增app一個 JavaScript 檔案



修改JS檔案編碼

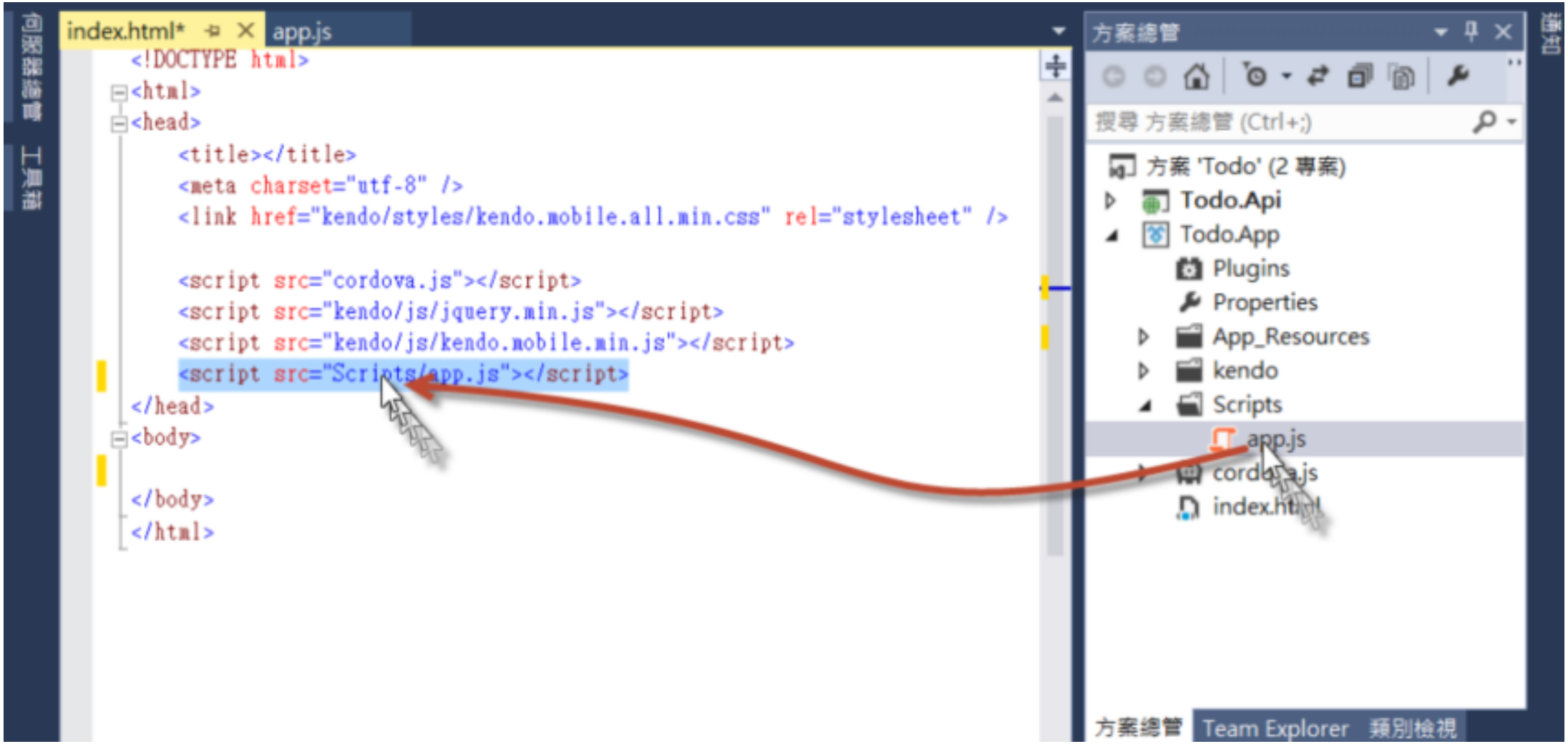

Todo - Microsoft Visual Studio (系統管理員)
檔案(F)
編輯(E)
檢視(V)
專案(P)
建置(B)
偵錯(D)
小組(M)
工具(T)
測試(S)
架構(C)
APPBUILDER
分析(N)
視窗(W)
Debug
Any CPU
開啟(O)
加入至原始檔控制
加入(D)
關閉(C)
關閉方案(T)
儲存 app.js(S) Ctrl+S
另存 app.js 為(A)...
1
2
另存新檔
Dropbox ▸ Demos ▸ Todo ▸ TodoApp ▸ Scripts
搜尋 Scripts
組合管理
新增資料夾
Microsoft Visual S
Projects
名稱
修改日期
類型
大小
app
2014/10/10 上午 ...
JS 檔案
0 KB
確認另存新檔
app.js 已經存在。
您要取代它嗎?
4
是(Y)
否(N)
進階儲存選項
C:\Users\keigen\Dropbox\Demos\Todo\TodoApp\Scripts\app.js
編碼方式(E):
Unicode (UTF-8 有簽章) - 字碼頁 65001
5
行尾結束符號(L):
目前的設定值
6
確定
取消
存檔(S)
取消
存檔(S)
以編碼方式儲存(V)...
3
100 %

接著加入程式碼

```
(function (global) {
    var app = global.app = global.app || {};

    app.application = new kendo.mobile.Application($(document.body));

    $(document).ajaxStart(function () {
        app.application.showLoading();
    });

    $(document).ajaxStop(function () {
        app.application.hideLoading();
    });

    app.callService = function (options) {
        $.ajax({
            url: options.url,
            type: options.requestType,
            data: options.data,
            dataType: options.dataType,
            // 如果有設定 HTTP 標頭，就將其加入
            beforeSend: function (xhr) {
                if (typeof options.httpHeader !== 'undefined' && typeof options.headerValue !== 'undefined') {
                    xhr.setRequestHeader(options.httpHeader, options.headerValue);
                }
            },
            // 成功時的回呼函式
            success: function (resultData) {
```

檢視（ View ）

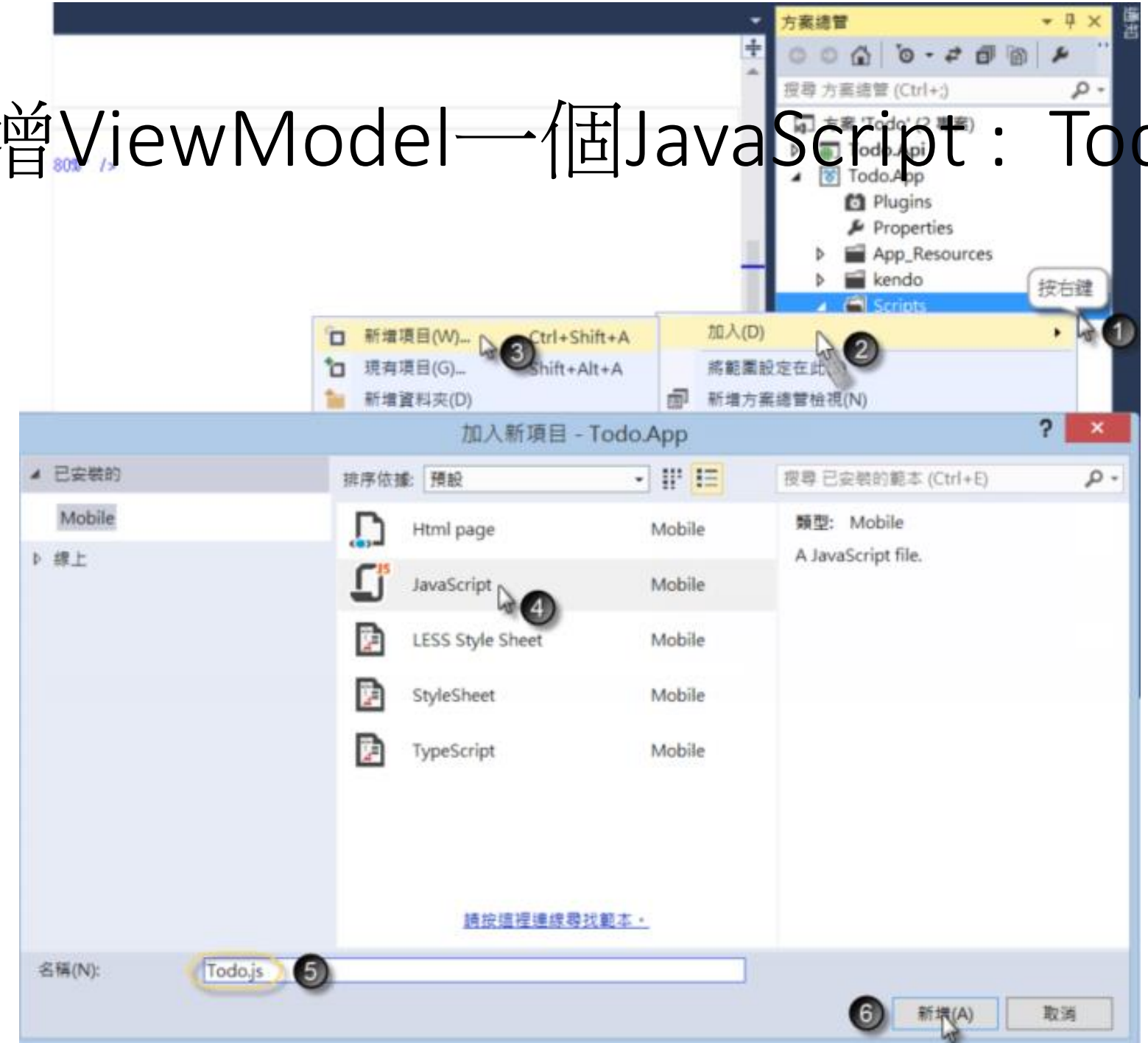
三個js拉進 HTML 檔案中參考



接著請body加下如下的程式碼：

```
<body>
    <div data-role="view"
        data-model="app.todo.viewModel"
        data-init="app.todo.init">
        <header data-role="header">
            <div data-role="navbar">
                <span data-role="view-title">待辦事項</span>
            </div>
        </header>
        <ul data-role="listview" data-style="inset">
            <li>
                <a data-bind="enabled: checkInput, click: onAdd" data-role="button" data-icon="add"></a>
                <input type="text" data-bind="value:description" placeholder="請輸入待辦事項" style="width:80%" />
            </li>
        </ul>
        <ul id=" todo" data-role="listview"
            data-template="todoListTemp"
            data-style="inset"
            data-bind="source:todoList"></ul>
    </div>
    <script type="text/x-kendo-template" id="todoListTemp">
        <button data-bind="click: onDelete, enabled:IsDone" data-role="button" data-icon="delete" />
        <span data-bind="text:Description"></span>
        <input type="checkbox" data-role="switch"
            data-bind="checked:IsDone ,events: { change: onChange }"
```

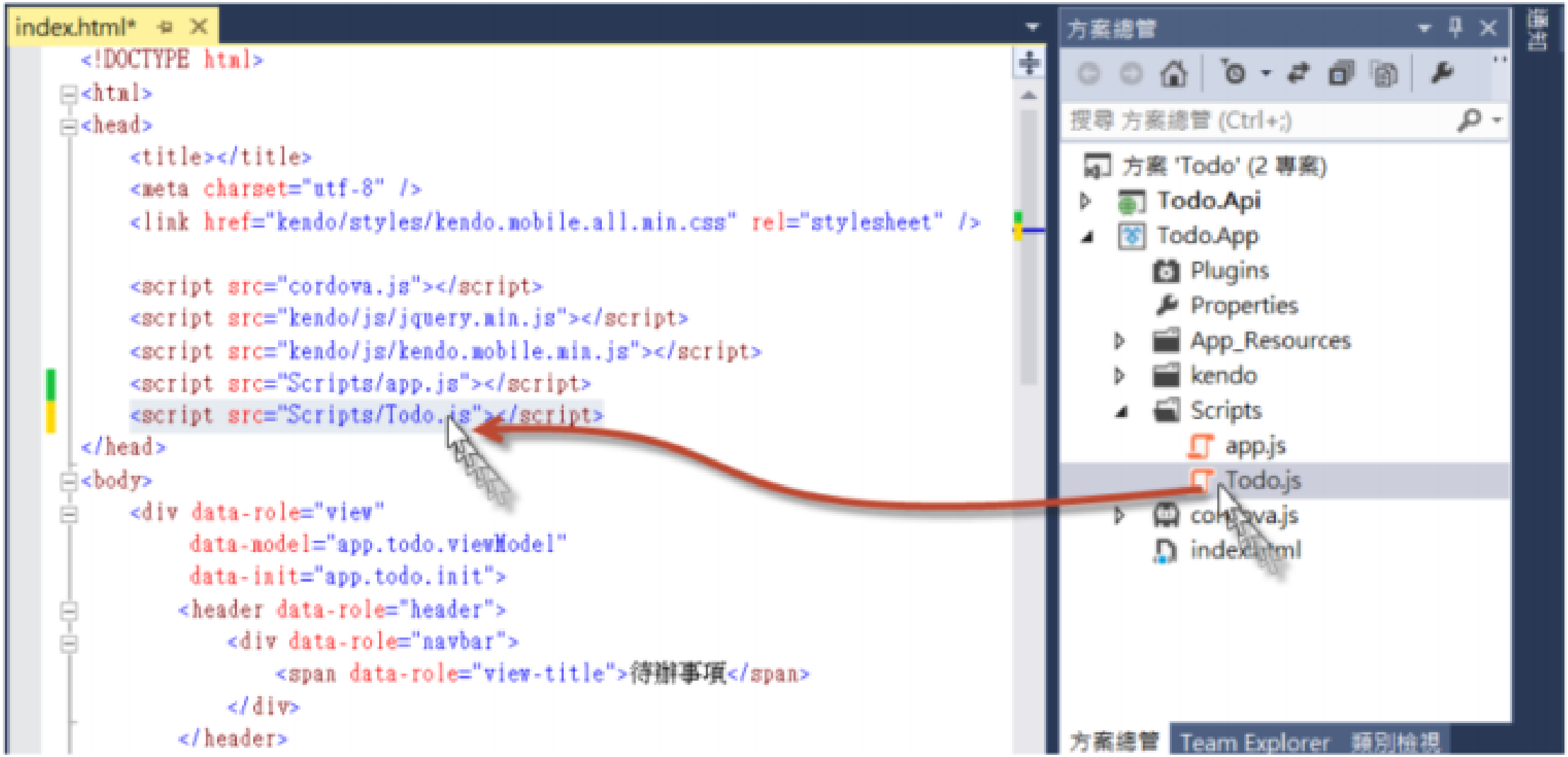
新增ViewModel一個JavaScript： Todo.js



接著加入程式碼

```
(function (global) {
    var app = global.app = global.app || {};
    var viewModel;

    viewModel = kendo.observable({
        description: "",
        todoList: [],
        getTodoList: function () {
            var options = {
                url: app.config.todoItemsUrl,
                requestType: "GET",
                dataType: "JSON",
                callBack: function (result) {
                    if (result.success === true) {
                        viewModel.set("todoList", result.data);
                    }
                }
            };
            app.callService(options);
        },
        onChange: function (e) {
            e.data.IsDone = e.checked;
            var options = {
                url: app.config.todoItemsUrl + e.data.Id,
                requestType: "PUT",
```

拉進 HTML 檔案中參考



補上三段 JavaScript 程式碼:app.js

- app. application = new kendo. mobile. Application($(document. body)) ;

- $(document) . aj axStart(function ( ) {

app. application. showLoading( ) ;

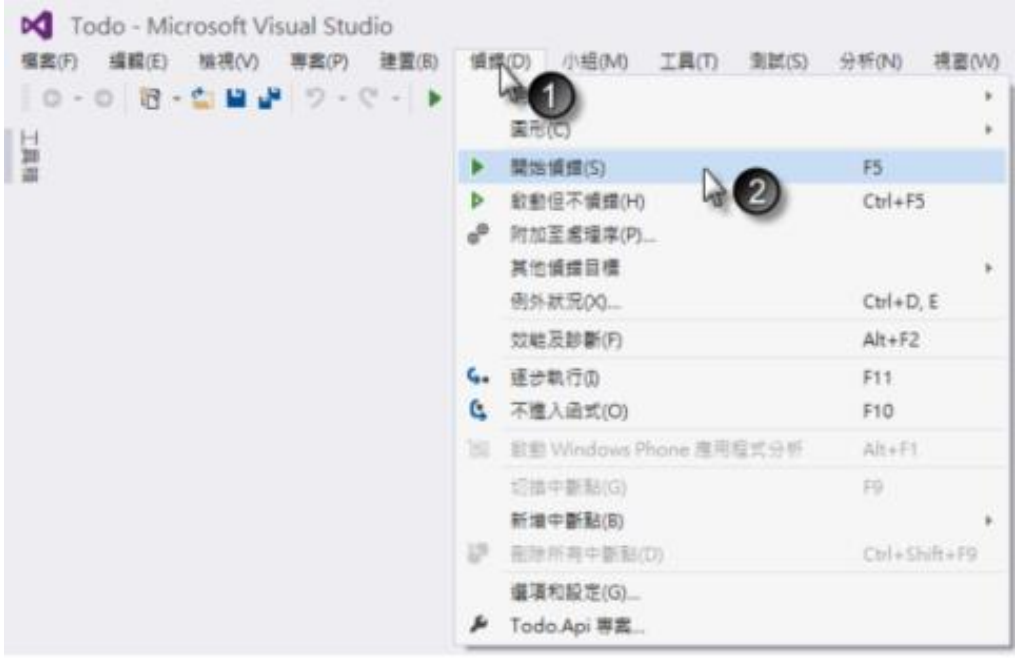
}) ;

- $(document) . aj axStop(function ( ) {

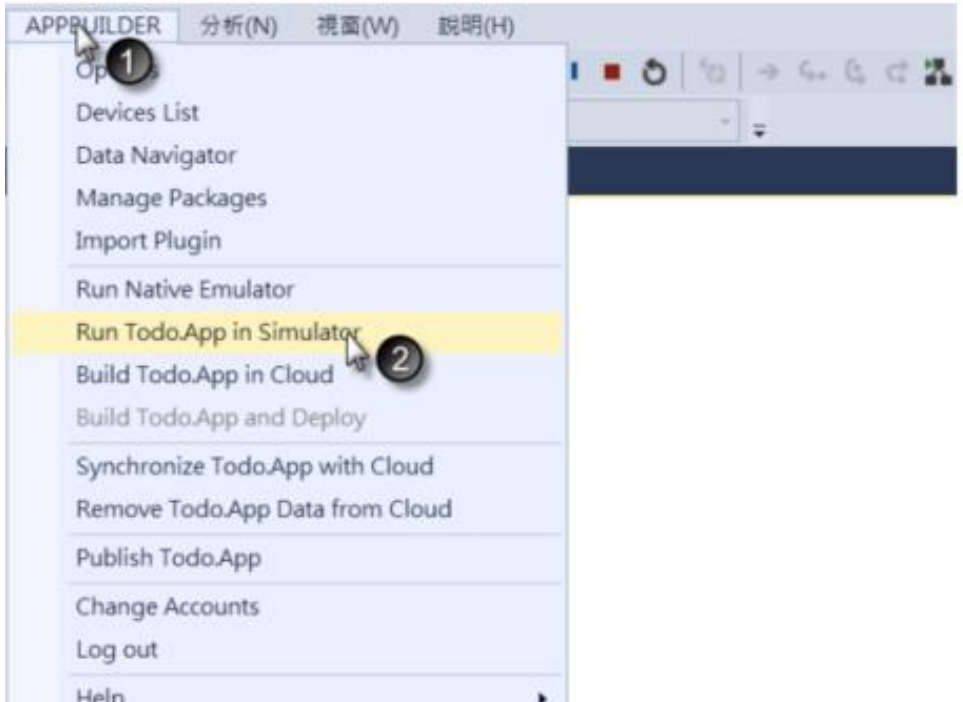
app. application. hideLoading( ) ;

}) ;

在模擬器中測試



接著，如圖所示開啟 AppBuilder 模擬器。



執行畫面:


Todo.App - Device Simulator
iPhone 6 Plus
8.0
Version
Flip
90° CCW
90° CW
Refresh
Reload
Screenshot
CARRIER
11:27 AM
100 %
待辦事項
請輸入待辦事項
買鹽酥雞
買肯德雞

Creating Help Pages for ASP.NET Web API

http://www.asp.net/web-api/overview/getting-started-with-aspnet-web-api/creating-api-help-pages