

.Net 蛤什麼?

.Net 博大精深，不是買本武功祕笈就能學成的 :P

2013-05-18

[ASP.NET] 使用 SSL Socket 建立 Client 與 Server 連接

👁 26051 💬 0 📄 ASP.NET (C#) 🕒 2013-05-31

請用「空白」區分關鍵字



[C#][ASP.NET][SSL][Socket]當 Client 與 Server 需要建立一個溝通的管道時可以使用 Socket 的方式建立一個通道，但是使用單純的 Socket 連線通道可能會擔心傳輸資料的過程中可能被截取修改因而不夠安全，為了防止這種情況我們可以使用建立 SSL Socket 的方式來進行資料的傳輸，所以這篇文章就來說明一下該如何建立 SSL Socket 通道，說實在本人對於憑證這個東西不是很熟悉，雖然在MSDN中已經有範例指導該如何建立 SSL Socket 方法，但是還是在憑證的操作上卡了一下，所以也會將卡住的地方舉出說明以免各位也卡在那兒。

前言

當 Client 與 Server 需要建立一個溝通的管道時可以使用 Socket 的方式建立一個通道，但是使用單純的 Socket 連線通道可能會擔心傳輸資料的過程中可能被截取修改因而不夠安全，為了防止這種情況我們可以使用建立 SSL Socket 的方式來進行資料的傳輸，所以這篇文章就來說明一下該如何建立 SSL Socket 通道，說實在本人對於憑證這個東西不是很熟悉，雖然在MSDN中已經有範例指導該如何建立 SSL Socket 方法，但是還是在憑證的操作上卡了一下，所以也會將卡住的地方舉出說明以免各位也卡在那兒。

範例將使用 SslStream類別 來說明建立的方法，SslStream 傳輸方式提供了訊息機密性和完整性檢查，當使用 SslStream 時可以防止傳輸的資訊被有心人讀取或竊改，使用 SslStream 時需要配合 TcpClient 與 TcpListener 一起使用，當客戶端需要與伺服器建立連線時需要提供X509憑證與服務器的X509憑證進行驗證，SSL 通訊協定協助為使用 SslStream 傳輸的訊息提供機密性和完整性檢查。當在用戶端和伺服器之間進行敏感資訊通訊時，應當使用 SSL 連接，例如由SslStream 提供的連接。使用 SslStream 可以協助防止任何人對在網路上傳輸的資訊進行讀取或竊改，在客戶端與服務器的憑證使用上差別在客戶端使用的憑證不需要包含私鑰 (*.cer)而伺服器則需要包含私鑰的憑證 (*.pfx)。

建立與使用憑證

建立憑證

首先要使用SSL就需要先建立一個憑證，但該如何建立跟使用呢? 可以先參考 使用憑證 此篇文章。

對於憑證的建立我們可以使用 Makecert.exe 工具，如果有裝 Visual Studio 則可以透過以下方式為建立，「開始」→「所有程式」→「Microsoft Visual Studio 2010」→「Visual Studio Tools」→「Visual Studio 命令提示字元 (2010)」

開啟命令提示字元後輸入：**makecert -r -pe -n "CN=SslSocket" -ss My -sky exchange**

參數說明如下：

- -r：建立自動簽名的憑證。
- -pe：將產生的私鑰金鑰標記為可匯出。如此可在憑證中加入私鑰金鑰。
- -n：指定主體的憑證名稱，使用雙引號包圍名稱開頭必須加CN=。
- -ss：指定主體的憑證存放區名稱，其儲存輸出憑證，My為憑證存放區的個人存放區。

作者



系列文章

- ASP.NET (C#) (57)
- ASP.NET (VB) (18)
- SQL (15)
- JavaScript (10)
- TypeScript (10)
- 其他資訊 (9)
- Web API (7)
- C# (7)
- MongoDB (5)
- Design Pattern (5)
- 自製工具 (2)
- Objective-C (2)



SslSocket.cer	2013/5/18 下午 0...	安全性憑證	1 KB
SslSocket.pfx	2013/5/18 下午 0...	個人資訊交換	2 KB

匯入憑證

憑證產生完成後就需要將產生的憑證匯入，SslSocket.pfx 之後將提供給服務器使用，而 SslSocket.cer 將提供給客戶端使用，憑證匯入的方式如下。

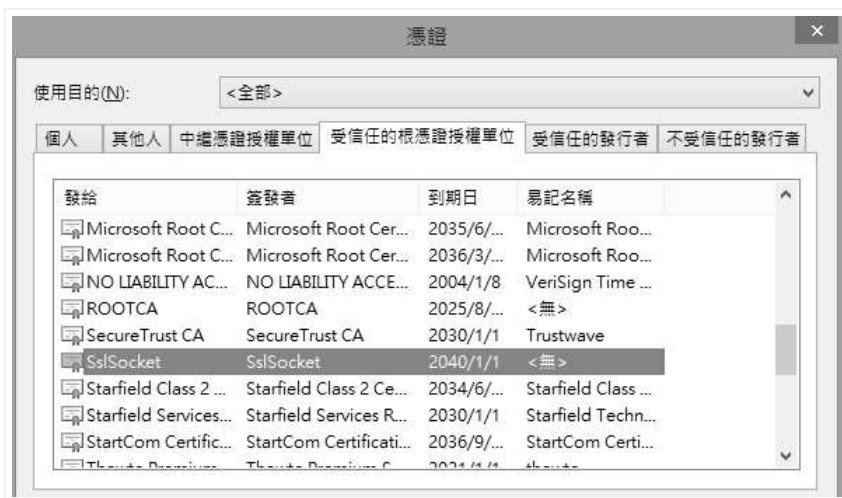
服務器憑證

將 SslSocket.pfx 放置在專案底下提供程式取用。

客戶端憑證

將 SslSocket.cer 於客戶端電腦使用以下步驟匯入。

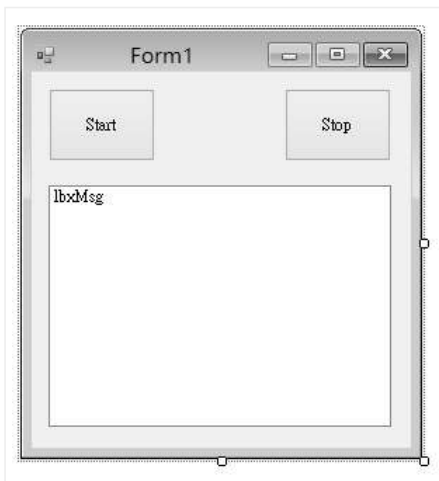
開啟「IE」→「工具」→「網際網路選項」→「內容」→「憑證」→「受信任的根憑證授權單位」→「匯入」



範例

Step 1

經過以上步驟後憑證的設定已經完成接下來就是要撰寫程式碼進行測試，首先建立一個 Windows Application 專案當作 Server 使用。



產生一個 SslSocket類別 加入以下程式碼

```
C#
public sealed class SslSocket
{
    private static TcpListener listener;
    private static X509Certificate ServerCertificate = null;
    private static bool IsRun = true;

    private static string _Certificate = string.Empty;
    public static string Certificate
    {
        get { return _Certificate; }
        set { _Certificate = value; }
    }

    /// <summary>
    /// 執行伺服器監聽
    /// </summary>
    public static void RunServer()
    {
        // 建立X509憑證
        ServerCertificate = new X509Certificate(Certificate, "ssl");
        // 監聽任何IP Address來的訊息
        listener = new TcpListener(System.Net.IPAddress.Any, 17170);
        // 開啟監聽
        listener.Start();

        while (IsRun)
        {
            UpdateStatus(string.Format("{0}-等待客戶端連接", DateTime.Now.ToString()));
            TcpClient client = listener.AcceptTcpClient();

            if (!IsRun)
            {
                listener.Stop();
                client.Close();
            }
            else
            {
                ProcessClient(client);
            }
        }
    }
}
```

```
/// <summary>
/// 停止服務器監聽
/// </summary>
public static void StopServer()
{
    IsRun = false;
    UpdateStatus(string.Format("{0}-停止客戶端連接", DateTime.Now.ToString("HH
}

/// <summary>
/// 接收客戶端訊息處理並回覆
/// </summary>
/// <param name="pClient"></param>
private static void ProcessClient(TcpClient pClient)
{
    SslStream sslStream = new SslStream(pClient.GetStream(), true);

    try
    {
        sslStream.AuthenticateAsServer(ServerCertificate, false, SslProtocol
        sslStream.ReadTimeout = 5000;
        sslStream.WriteTimeout = 5000;
        UpdateStatus(string.Format("{0}-等待客戶端訊息", DateTime.Now.ToString

        string messageData = ReadMessage(sslStream);

        UpdateStatus(string.Format("{0}-接收訊息內容: {1}", DateTime.Now.ToStr
        byte[] message = Encoding.UTF8.GetBytes(string.Format("服務器已接收此:

        UpdateStatus(string.Format("{0}-回覆客戶端訊息", DateTime.Now.ToString
        sslStream.Write(message);
    }
    catch (Exception)
    {
        sslStream.Close();
        pClient.Close();
        return;
    }
    finally
    {
        sslStream.Close();
        pClient.Close();
    }
}

/// <summary>
/// 讀取訊息內容
/// </summary>
/// <param name="pSslStream"></param>
/// <returns></returns>
private static string ReadMessage(SslStream pSslStream)
{
    byte[] buffer = new byte[2048];
    StringBuilder messageData = new StringBuilder();
    int bytes = -1;
    do
    {
        bytes = pSslStream.Read(buffer, 0, buffer.Length);

        Decoder decoder = Encoding.UTF8.GetDecoder();
        char[] chars = new char[decoder.GetCharCount(buffer, 0, bytes)];
        decoder.GetChars(buffer, 0, bytes, chars, 0);
```

```
        if (messageData.ToString().IndexOf("<EOF>") != -1)
        {
            break;
        }
    } while (bytes != 0);

    return messageData.ToString();
}

/// <summary>
/// 更新主視窗ListBoxUI
/// </summary>
/// <param name="pMessage"></param>
private static void UpdateStatus(string pMessage)
{
    Form1.MainListBox.Invoke(new Action(() => Form1.MainListBox.Items.Add(pM
}
}
```

此類別中 RunServer 方法為啟動監聽需呼叫的方法，首先將透過 X509Certificate 建構函式 (String, String) 建立一個 X509 憑證存入金鑰容器，傳入參數為 (FileName, Password)，之後建立 TcpListener 物件用來監聽來至於 TCP 客戶端的連結且在此需要指定本機 IP 及 Port，而在 While 迴圈內則建立 TcpClient 物件取得客戶端來連接時的 NetworkStream 的資料流，在此 TcpListener 使用了 AcceptTcpClient (接受暫止連線要求) 方法，意思是此監聽將處於暫時靜止狀態，當客戶端有連結時才會回應，所以最好將 SslSocket 類別使用執行序執行以避免主線程阻塞。

當客戶端已連接後會執行 ProcessClient 方法，此時將建立 SslStream 物件來接收客戶端傳送來的資料流並且進行服務器的憑證驗證，而後進行資料的讀取動作，其中<EOF>標籤為標註訊息的結尾判斷使用，最後將處理完的資料回入資料流中傳送至客戶端處理。

接著在表單程式中產生一個執行序去執行 RunServer 方法啟動監聽同時還需要指定服務器使用的憑證。

```
C#
public partial class Form1 : Form
{
    public static ListBox MainListBox;

    public Form1()
    {
        InitializeComponent();

        MainListBox = this.lbxMsg;
    }

    private void btnStart_Click(object sender, EventArgs e)
    {
        Thread socket = new Thread(RunSocket);
        socket.IsBackground = true;
        socket.Start();
    }

    private void btnStop_Click(object sender, EventArgs e)
    {
        SslSocket.StopServer();
    }

    private void RunSocket()
    {
        SslSocket.Certificate = Application.StartupPath + @"\SslSocket.pfx";
        SslSocket.RunServer();
    }
}
```

本頁段落

- 建立憑證
- 匯出與匯入憑證
- Step 1
- Step 2
- 測試結果





Step 2

接下來建立一個客戶端用來連接服務器溝通，建立一個 Web 網站於方案中，將剛剛產生的 SslSocket.cer 憑證放置在網站底下，簡單拉一個測試畫面。



建立一個 SendToServer 類別，主要工作於將客戶端訊息傳送至服務器端，詳細代碼如下。

```

C#
public class SendToServer
{
    public string HostAddress { get; set; }
    public int HostPort { get; set; }

    /// <summary>
    /// 建構子，傳入服務器IP及Port
    /// </summary>
    /// <param name="pHostAddress"></param>
    /// <param name="pHostPort"></param>
    public SendToServer(string pHostAddress, int pHostPort)
    {
        HostAddress = pHostAddress;
        HostPort = pHostPort;
    }

    /// <summary>
    /// 執行將訊息發送至服務器方法
    /// </summary>
    /// <param name="pMessage"></param>
    /// <returns></returns>
    public string SendMsgToServer(string pMessage)
    {
        TcpClient client = new TcpClient(HostAddress, HostPort);
        SslStream sslStream = new SslStream(client.GetStream(), false, new RemoteCertificateValidationCallback(delegate { return true; }));
        X509CertificateCollection certs = new X509CertificateCollection();
        X509Certificate cert = X509Certificate.CreateFromCertFile(HttpContext.Current.Server.MapPath("~/certs/SslSocket.cer"));
        certs.Add(cert);
        try
        {
            sslStream.AuthenticateAsClient("SslSocket", certs, System.Security.Authentication.SslProtocols.Tls12);
        }
        catch (Exception ex)
        {
            client.Close();
            return ex.Message;
        }
        byte[] message = Encoding.UTF8.GetBytes(string.Format("{0}<EOF>", pMessage));
        sslStream.Write(message);
        sslStream.Flush();
        string serverMessage = ReadMessage(sslStream);
        client.Close();
        return serverMessage;
    }
}

```

```
/// <summary>
/// 讀取訊息內容
/// </summary>
/// <param name="pSslStream"></param>
/// <returns></returns>
private string ReadMessage(SslStream pSslStream)
{
    byte[] buffer = new byte[2048];
    StringBuilder messageData = new StringBuilder();
    int bytes = -1;
    do
    {
        bytes = pSslStream.Read(buffer, 0, buffer.Length);

        Decoder decoder = Encoding.UTF8.GetDecoder();
        char[] chars = new char[decoder.GetCharCount(buffer, 0, bytes)];
        decoder.GetChars(buffer, 0, bytes, chars, 0);
        messageData.Append(chars);

        if (messageData.ToString().IndexOf("<EOF>") != -1)
        {
            break;
        }
    } while (bytes != 0);

    return messageData.ToString();
}

/// <summary>
/// 驗證服務器SSL憑證
/// </summary>
/// <param name="sender"></param>
/// <param name="certificate"></param>
/// <param name="chain"></param>
/// <param name="sslPolicyErrors"></param>
/// <returns></returns>
public static bool ValidateServerCertificate(object sender, X509Certificate
{
    if (sslPolicyErrors == SslPolicyErrors.None)
        return true;
    return false;
}
}
```

在 SendMsgToServer 方法中，開始先建立 TcpClient 物件連接至指定的服務器 IP 及 Port，之後建立 SslStream 物件並透過 AuthenticateAsClient 方法驗證與服務器的憑證，驗證成功後將訊息轉換成 byte[] 寫入資料流中傳送至服務器處理，服務器處理完成後將回寫資料至客戶端進行解析後顯示。

最後在測試頁面程式碼中，建立一個 SendToServer 物件並且指定其 IP 及 Port 後呼叫 SendMsgToServer 方法即可。

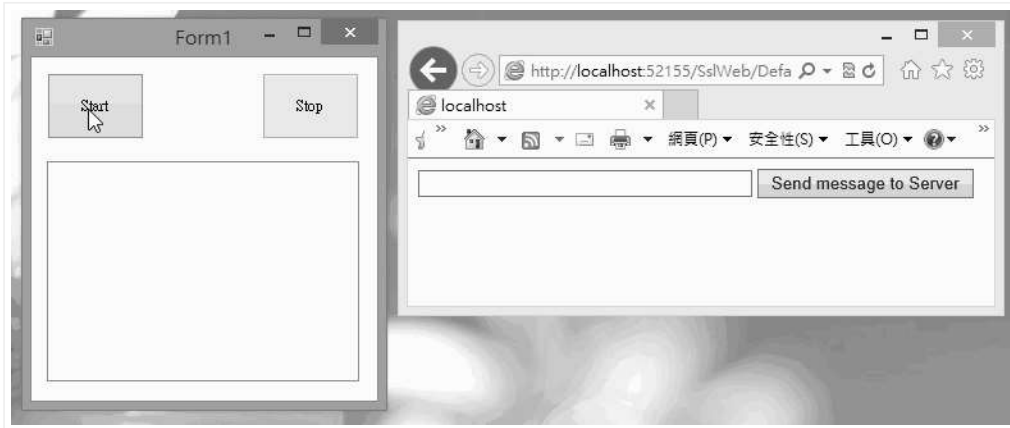
```
C#
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void btnSend_Click(object sender, EventArgs e)
    {
        SendToServer send = new SendToServer("127.0.0.1", 17170);
    }
}
```



```
}  
}
```

測試結果



以上為參考MSDN範例而產生的一個簡單的使用說明。

範例程式碼

TSocket.rar

參考資料

SslStream 類別

X509Certificate 類別

TcpListener 類別

TcpClient 類別

NetworkStream 類別

以上文章敘述如有錯誤及觀念不正確，請不吝指教
如有侵權內容也請您與我反應~謝謝您:)

ASP.NET · Socket · SSL

[回首頁](#)

關聯文章

[ASP.NET] 在 Server 接收 Client 回傳的 JSON 格式資料

[VB] ASP.NET 計算線上人數方法

[ASP.NET] 使用 Google Analytics 並整合 Google Analytics 流量分析資料於網站中顯示

[ASP.NET] TypeScript + AngularJs App 初始化撰寫方式

[ASP.NET] 自訂網站 Exception 錯誤導向處理

[ASP.NET] 使用Lock鎖定 在多使用者或多執行緒下新增編號確保編號不重複

[VB][jQuery] 使用ajax在原頁面上加載一個新頁面

[ASP.NET] 使用 Web Service Redirection

[ASP.NET][JS&jQuery] Client取得Server日期時間

[ASP.NET] TypeScript + AngularJs 自訂 ErrorHandlerProvider

[.NET] 在 Debug 或 Release 時執行特定程式碼

[ASP.NET] 圖形驗證碼破解-以簡單圖形為例

[ASP.NET] 於ASP.NET將Facebook帳號整合

[ASP.NET] 解決在 User Control 中使用 Bundling 發生重複輸出 Script 或 Style 問題

[VB][jQuery][Ajax] 使用Ajax json返回Server取值的方法

[ASP.NET] 使用 Web Deployment Projects 部署 ASP.NET 網站

[VB] 使用圖形驗證碼範例

[ASP.NET] TypeScript + AngularJs 系列 - 目錄

[ASP.NET] 使用 HttpRequest POST/GET 方法 (解決錯誤417,Server重新導向)

[ASP.NET][VB][C#] 在同一個網站內使用 VB / C# 類別

[ASP.NET] 將不同貨幣格式字串轉型至 Decimal 型別

[ASP.NET] 解決大量發送 HttpRequest 產生 Timeout 錯誤問題

[ASP.NET] TypeScript + AngularJs Directive 撰寫方式

[VB] VB內建函式

[ASP.NET] 使用 OAuth 2.0 整合 Google、Facebook、WindowsLive、Xuite 登入

[ASP.NET] TypeScript + AngularJs 使用 ASP.NET Bundle 與自動判斷載入 JS

[ASP.NET] 使用NPOI產生EXCEL

[ASP.NET] 使用 Global 打造背景批次作業

[VB][jQuery] 使用 jQuery.autocomplete 實現自動完成文字框

[VB][jQuery] 使用 jQuery.uploadify 上傳檔案

[ASP.NET] 使用 ODBC 匯入 CSV 資料

[VB][XML] 將資料庫資料寫入XML再使用GridView讀出

[C#] 於 Socket 連線中使用非同步處理作業

[ASP.NET] TypeScript + AngularJs Controller 撰寫方式

[ASP.NET] ASP.NET 常見函數型態

[ASP.NET] TypeScript + AngularJs Service 撰寫方式

[VB] DataTable 不使用 Compute 做 SUM 運算

[ASP.NET] jQuery UI MultiSelect Widget 下拉多選框

[ASP.NET][VB][C#] 日期格式速查表

[ASP.NET] 解決 jQuery 配 UpdatePanel 後第二次執行失效問題

[ASP.NET] 啟動 SignalR 發生 System.Reflection.TargetInvocationException 例外

[JavaScript&jQuery] 讓GridView可以使用全選/取消的CheckBox功能

[ASP.NET] 解決當使用 FriendlyUrls 網站執行 ajax 呼叫 WebMethod 錯誤

[VB] 圖片等比例縮小方法

[VB] 在MasterPage下使用FindControl找到控制項

[.NET] IoC (Inversion of Control) 實作工廠模式

[ASP.NET] Application、Page、Control 生命週期

[ASP.NET] 解決泛型處理常式 (Handler.ashx) 於 IIS Express 使用 HTTP PUT、DELETE 動詞 405 Method Not Allowed 錯誤

[ASP.NET] 使用 SqlBulkCopy 批次新增大量資料

[VB][C#] 解決WebUserControl/ASCX 在網站發行時出現 'XXXXX' 模稜兩可的錯誤

[ASP.NET] TypeScript + AngularJs 專案架構

[ASP.NET] Boxing(裝箱) 與 UnBoxing(拆箱) 測試

[ASP.NET][LINQ] 基本LINQ語法介紹

[ASP.NET] 在C#中模擬VB幾個常用的函數

[ASP.NET] 處理 Cross Domain Callback 跨網域呼叫方法

[ASP.NET] 使用 OleDb 匯入 Excel 資料

[ASP.NET] 利用反射方法(Type.InvokeMember)取得不同資料物件屬性

[ASP.NET] 判斷使用者是電腦還是手機瀏覽本網頁

[VB] 將讀出的資料去除HTML標籤方法

[徵才] 本公司招募程式設計師 2-3 名

[ASP.NET] 實現與防範 CSRF 跨網站請求偽造攻擊

[C#][VB] 使用 UserControl 建立事件於 Page 頁面調用方法

[ASP.NET] 使用 DotNetZip Library 壓縮及解壓Zip檔案

[ASP.NET] TypeScript 自訂第三方套件 DefinitelyTyped

[VB][jQuery] 使用 ajax+ashx+json 做下拉選單連動效果(DropDownList Ajax)

[ASP.NET] Bundling Script and Style 加速網站效能

[ASP.NET] 自動啟用相容性檢視? 內部網路網站設定在搞鬼

[ASP.NET] 使用 HttpUtility.ParseQueryString 剖析網址參數

[.NET] 實作工廠模式、IoC (Inversion of Control) 與 DI (Dependency Injection)

系列文章

[C#] Datatable.Select 的運算式用法

[VB][C#] 取得 Request URL 方法

[VB][C#] 使網站開啟相容性檢視方法

[VB][C#] 使用 Url Rewriter 套件隱藏網址參數

[VB][C#] 解決WebUserControl/ASCX 在網站發行時出現 'XXXXX' 模稜兩可的錯誤

[ASP.NET][VB][C#] 在同一個網站內使用 VB / C# 類別

[VB.NET] 命令列引數 Environment 使用方法

[ASP.NET] Session 遺失 / Session Timeout / Session 設定

[ASP.NET] Application、Page、Control 生命週期

[ASP.NET][VB][C#] 日期格式速查表

[ASP.NET][JS&jQuery] Client取得Server日期時間

[C#][VB] 使用 UserControl 建立事件於 Page 頁面調用方法

[ASP.NET] 使用 Web Deployment Projects 部署 ASP.NET 網站

[ASP.NET] 於ASP.NET使用 jQuery EasyUi DataGrid 存取資料

[ASP.NET] 如何從Server端將資料轉換成JSON格式傳至Client端

[ASP.NET] 於ASP.NET將Facebook帳號整合

[ASP.NET] 使用 Google Analytics 並整合Google Analytics 流量分析資料於網站中顯示

[ASP.NET] 使用Lock鎖定 在多使用者或多執行緒下新增編號確保編號不重複

[ASP.NET] 判斷使用者是電腦還是手機瀏覽本網頁

[ASP.NET] 在C#中模擬VB幾個常用的函數

[ASP.NET] 使用 DotNetZip Library 壓縮及解壓Zip檔案

[ASP.NET][LINQ] 基本LINQ語法介紹

[ASP.NET] 使用 Log4net 記錄網站訊息

[.NET] IoC (Inversion of Control) 實作工廠模式

[ASP.NET] 在 Server 接收 Client 回傳的 JSON 格式資料

[ASP.NET] 使用 OAuth 2.0 整合 Google、Facebok、WindowsLive、Xuite 登入

[ASP.NET] 使用 OleDb 匯入 Excel 資料

[ASP.NET] 解決 jQuery 配 UpdatePanel 後第二次執行失效問題

[ASP.NET] 使用 ODBC 匯入 CSV 資料

[.NET] 實作工廠模式、IoC (Inversion of Control) 與 DI (Depedency Injection)

[ASP.NET] 使用 configSections 自訂組態區段(web.config/app.config)

[ASP.NET] 使用 HttpRequest POST/GET 方法 (解決錯誤417,Server重新導向)

[ASP.NET] 為 GridView 加上流水號

[ASP.NET] 使用 Global 打造背景批次作業

[ASP.NET] 使用 SqlBulkCopy 批次新增大量資料

[TFS] 於 TFS 強制簽入已被簽出的檔案

[ASP.NET] 使用NPOI產生EXCEL

[資安] Input 輸入元素應避免使用 Disabled 屬性與直接取得元素所選擇的值

[ASP.NET] 自訂 GridView PagerTemplate 分頁

[ASP.NET] 使用 Web Service Redirection

[ASP.NET] 使用 SSL Socket 建立 Client 與 Server 連接

[ASP.NET] 自訂網站 Exception 錯誤導向處理

[ASP.NET] 使用 HttpUtility.ParseQueryString 剖析網址參數

[ASP.NET] 圖形驗證碼破解-以簡單圖形為例

[ASP.NET] 將不同貨幣格式字串轉型至 Decimal 型別

[ASP.NET] Web Service 與 WCF 差異表筆記

[ASP.NET] Boxing(裝箱) 與 UnBoxing(拆箱) 測試

[ASP.NET] 實現與防範 CSRF 跨網站請求偽造攻擊

[ASP.NET] 解決泛型處理常式 (Handler.ashx) 於 IIS Express 使用 HTTP PUT、DELETE 動詞 405 Method Not Allowed 錯誤

[ASP.NET] Bundling Script and Style 加速網站效能

[ASP.NET] 解決當使用 FriendlyUrls 網站執行 ajax 呼叫 WebMethod 錯誤

[ASP.NET] 解決大量發送 HttpRequest 產生 Timeout 錯誤問題

[ASP.NET] 解決在 User Control 中使用 Bundling 發生重複輸出 Script 或 Style 問題

[ASP.NET] 關閉 HTML5 表單驗證功能

[ASP.NET] 自動啟用相容性檢視？內部網路網站設定在搞鬼

Sponsored

立即打開您的瀏覽器進入遊戲

Hero Wars

一張家庭照片破壞了和平。妻子提出離婚

The Family Breeze

毛孩口臭、卻不給刷牙？日本流行這樣做

嚼嚼口腔的健康革命

[了解更多](#)

經典歷史時刻的45張不尋常的照片

2FunNews

白開水+它，睡前喝一杯，脂肪瘋狂燃燒一整晚，輕鬆瘦出馬甲線（今晚試試）

日本新谷酵素黃金版專賣

臺灣最年輕美貌的15位女星

BestFamilyMag

20 張奇怪的照片在我們仔細觀察之前看起來很普通

The Travel Breeze

Sponsored

立即打開您的瀏覽器進入遊戲

Hero Wars

一張家庭照片破壞了和平。妻子提出離婚

The Family Breeze

毛孩口臭、卻不給刷牙？日本流行這樣做

嚼嚼口腔的健康革命

了解更多

白開水+它，睡前喝一杯，脂肪瘋狂燃燒一整晚，輕鬆瘦出馬甲線（今晚試試）

日本新谷酵素黃金版專賣

經典歷史時刻的45張不尋常的照片

2FunNews

20 張奇怪的照片在我們仔細觀察之前看起來很普通

The Travel Breeze