黑暗執行緒
DARKTHREAD

MicroHttpServer - 用100行C#寫一個HTTP Server

2010-08-14 09:23 AM 19 49,156

有個點子，想在WinForm上跑程式模擬出Web Server功能，讓Browser或程式可以透過HTTP協定與其溝通。既然想到，就動手做看看囉!

HTTP Server絕大部分的核心功能，其實都可用.NET搞定: 用TcpListener <http://msdn.microsoft.com/en-us/library/system.net.sockets.tcplistener.aspx?WT.mc_id=DOP-MVP-37580> 接受特定Port連入的TCP連線，取得NetworkStream <http://msdn.microsoft.com/en-us/library/system.net.sockets.networkstream.aspx?WT.mc_id=DOP-MVP-37580> ，以StreamReader、StreamWriter讀取及寫入資料... .NET BCL <http://en.wikipedia.org/wiki/Base_Class_Library> 真是應有盡有!相較之下，以前那種基礎元件跟函式庫都得自己張羅的時代，只能用茹毛飲血來形容。

有了BCL的加持，配合兩個自訂類別封裝Request、Response，只花了不到100行C#，就組出一個可以接受HTTP Request，傳回結果的超迷你HTTP Server!

排版顯示 純文字

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Threading;

namespace DarkHttpServer
{
    //Reuquest物件
    public class CompactRequest
    {
        public string Method, Url, Protocol;
        public Dictionary<string, string> Headers;
        //傳入StreamReader，讀取Request傳入的內容
        public CompactRequest(StreamReader sr)
        {
            //第一列格式如: GET /index.html HTTP/1.1
            string firstLine = sr.ReadLine();
            string[] p = firstLine.Split(' ');
            Method = p[0];
            Url = (p.Length > 1) ? p[1] : "NA";
            Protocol = (p.Length > 2) ? p[2] : "NA";
            //讀取其他Header，格式為HeaderName: HeaderValue
            string line = null;
            Headers = new Dictionary<string, string>();
            while (!string.IsNullOrEmpty(line = sr.ReadLine()))
            {
                int pos = line.IndexOf(":");
                if (pos > -1)
                    Headers.Add(line.Substring(0, pos),
                        line.Substring(pos + 1));
            }
        }
    }
    //Response物件
    public class CompactResponse
    {
        //預設200, 404, 500三種回應
        public class HttpStatus
        {
            public static string Http200 = "200 OK";
            public static string Http404 = "404 Not Found";
            public static string Http500 = "500 Error";
        }
        public string StatusText = HttpStatus.Http200;
        public string ContentType = "text/plain";
        //可回傳Response Header
        public Dictionary<string, string> Headers
            = new Dictionary<string, string>();
        //傳回內容，以byte[]表示
        public byte[] Data = new byte[] { };
    }
    //簡陋但堪用的HTTP Server
    public class MicroHttpServer
    {
        private Thread serverThread;
        TcpListener listener;
        //呼叫端要準備一個函數，接收CompactRequest，回傳CompactResponse
        public MicroHttpServer(int port,
            Func<CompactRequest, CompactResponse> reqProc)
        {
            IPAddress ipAddr = IPAddress.Parse("127.0.0.1");
            listener = new TcpListener(ipAddr, port);
            //另建Thread執行
            serverThread = new Thread(() =>
            {
                listener.Start();
                while (true)
                {
                    Socket s = listener.AcceptSocket();
                    NetworkStream ns = new NetworkStream(s);
                    //解讀Request內容
                    StreamReader sr = new StreamReader(ns);
                    CompactRequest req = new CompactRequest(sr);
                    //呼叫自訂的處理邏輯，得到要回傳的Response
```

```
                    CompactResponse resp = reqProc(req);
                    //傳回Response
                    StreamWriter sw = new StreamWriter(ns);
                    sw.WriteLine("HTTP/1.1 {0}", resp.StatusText);
                    sw.WriteLine("Content-Type: " + resp.ContentType);
                    foreach (string k in resp.Headers.Keys)
                        sw.WriteLine("{0}: {1}", k, resp.Headers[k]);
                    sw.WriteLine("Content-Length: {0}", resp.Data.Length);
                    sw.WriteLine();
                    sw.Flush();
                    //寫入資料本體
                    s.Send(resp.Data);
                    //結束連線
                    s.Shutdown(SocketShutdown.Both);
                    ns.Close();
                }
            });
            serverThread.Start();
        }
        public void Stop()
        {
            listener.Stop();
            serverThread.Abort();
        }
    }
```

```
}
```

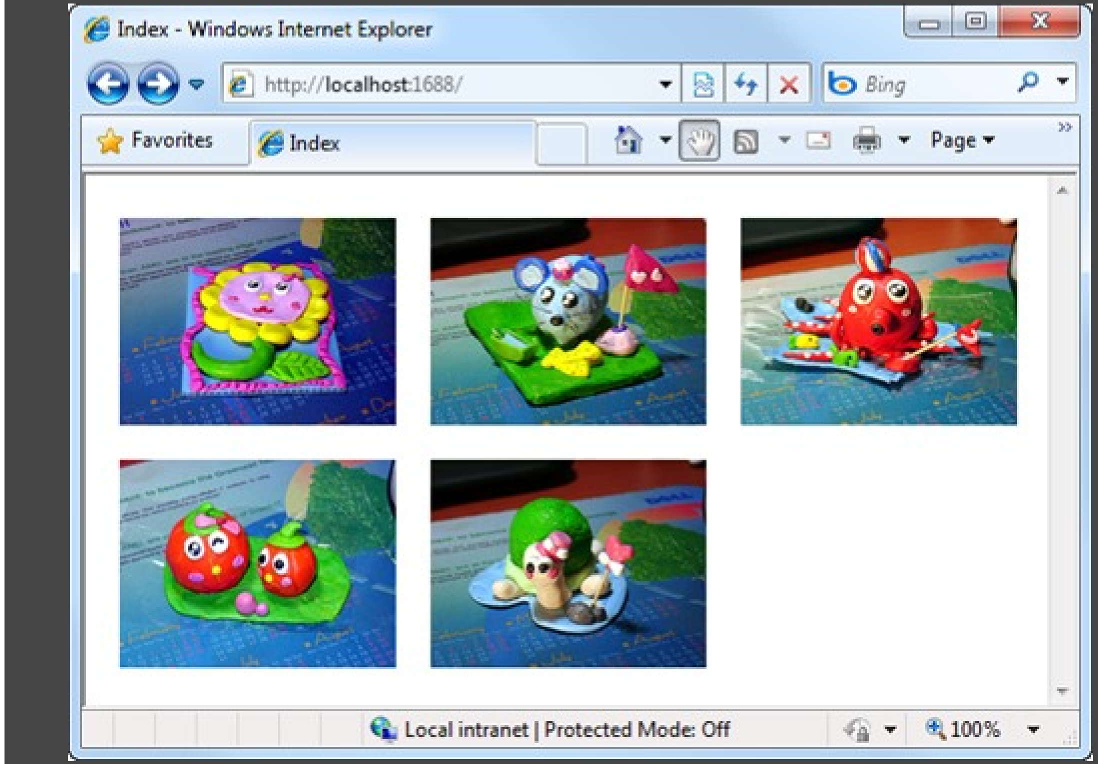
好了，有了MicroHttpServer類別，我們來寫一個小小的Console Application，做一個將特定目錄下JPG圖檔以網頁方式呈現的迷你Web Server當作應用範例:

排版顯示 純文字

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;

namespace DarkHttpServer
{
    class Program
    {
        static string path = @"C:\temp\arts";

        static void Main(string[] args)
        {
            MicroHttpServer mhs = new MicroHttpServer(1688,
            (req) =>
            {
                if (req.Url == "/")
                    return ListPhoto(req);
                else if (req.Url.EndsWith(".jpg"))
```

```
                    return GetJpeg(
                        Path.Combine(path, req.Url.TrimStart('/')));
                else return new CompactResponse()
                {
                    StatusText = CompactResponse.HttpStatus.Http500
                };
            });
            Console.Write("Press any key to stop...");
            Console.Read();
            mhs.Stop();
        }

        //列出圖檔，組成網頁傳回
        static CompactResponse ListPhoto(CompactRequest req)
        {
            StringBuilder sb = new StringBuilder();
            sb.Append(@"
<html><head><title>Index</title>
<style type='text/css'>img {
    width: 160px; height: 120px; float: left;
    margin: 10px;
}</style>
</head><body>
");
            sb.Append("");
            foreach (string file in
                Directory.GetFiles(path, "*.jpg"))
                sb.AppendFormat("<img src='{0}' />",
                    Path.GetFileName(file));
            sb.Append("</body></html>");
            return new CompactResponse()
            {
                ContentType = "text/html",
                Data = Encoding.UTF8.GetBytes(sb.ToString())
            };
        }
        //取得圖檔
        static CompactResponse GetJpeg(string file)
        {
            if (File.Exists(file))
                return new CompactResponse()
                {
                    ContentType = "image/jpeg",
                    Data = File.ReadAllBytes(file)
                };
            else //找不到檔案時傳回HTTP 404
                return new CompactResponse()
                {
                    StatusText = CompactResponse.HttpStatus.Http404
                };
        }
    }
}
```

用IE連上httq://localhost:1688，薑薑薑薑! 小閃光的[手工藝品 <http://blog.darkthread.net/blogs/darkthreadtw/archive/2010/08/07/2010-summer-arts.aspx>](http://blog.darkthread.net/blogs/darkthreadtw/archive/2010/08/07/2010-summer-arts.aspx)在他爹的"手工藝品"上被展示出來了。(這證明我的手也很巧呀! XD)


Index - Windows Internet Explorer
http://localhost:1688/
Bing
Favorites
Index
Page
Local intranet | Protected Mode: Off
100%

請大家跟我一起高呼: .NET好威呀!

惡補深度學習的數學基礎
打好 Python 與常用套件

讚 21 人說這個讚。趕快註冊來看看朋友對哪些內容按讚。