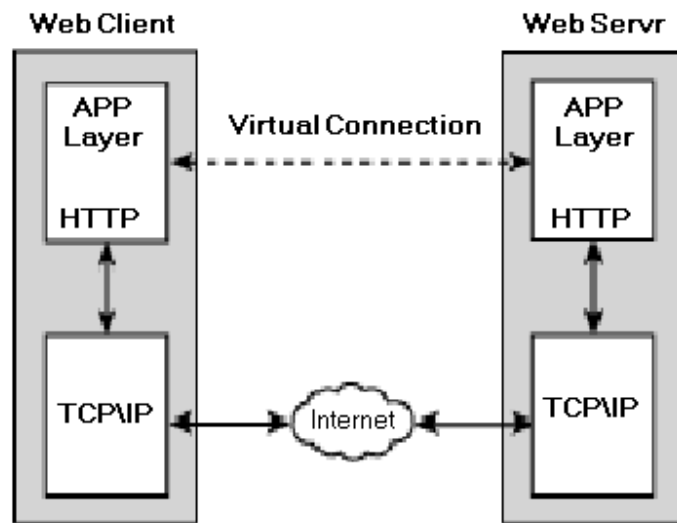


الدرس السابع: HTTP – Hyper Text Transfer Protocol Programming

تتلخص وظيفة ال HTTP بشكل عام على انه البرتوكول المستخدم لتوصيل طلب المستخدم User Request إلى الويب سيرفر ثم قيام ال web server بالرد على ال Request والذي يسمى ب Server Response وتأكيد تستطيع نقل جميع أشكال ال (Multimedia) من النص وصورة و صوت و فيديو وغيره .. من ال Web Server إلى ال Client Application باستخدام Byte Stream object.

يعمل برتوكول ال HTTP على ال Application Layer وهذا يعني استخدامه بشكل مباشر من واجهة المستخدم كما هو الحال في FTP,POP3,SMTP,DNS انظر إلى الشكل التالي:



أولا : Downloading From Web Server

نستطيع التعامل مع ال Web Server في الدوت نيت باستخدام الكلاس WebClient الموجود في النيم سبيس System.Net إذ تقدم لنا جميع الإمكانيات لتوصيل طلب الزبون و الرد عليه Server Response & User Request وتدعم ال WebClient Class ثلاثة Methods لتحميل البيانات من ال Web Server وهي:

1- **DownloadData** ووظيفتها جلب البيانات من ال Web Server وتخزينها في Byte Array وتعرض على شكل HTML Code وتستخدم كما يلي كمثال :

```
using System;
using System.Net;
using System.Text;
class DownloadData_Method
{
    public static void Main ()
    {
        WebClient wc = new WebClient();
        byte[] response = wc.DownloadData("http://www.google.com");
        Console.WriteLine(Encoding.ASCII.GetString(response));
    }
}
```

DownloadFile -2 ووظيفتها نقل ملف ما من ال Web Server وتخزينها مباشرة في Local Computer وهو سهل الاستخدام جدا إذ ما عليك سوا تمرير موقع الملف والمكان الذي تريد تخزين الملف فيه ويستخدم كما يلي كمثال :

```
using System;
using System.Net;

class DownloadFile_Method
{
    public static void Main ()
    {
        WebClient wc = new WebClient();
        string filename = "C:\\ra.zip";

        Console.WriteLine("Download in Progress Please Waite...");

        wc.DownloadFile("http://www.personalmicrocosms.com/zip/ra.zip", filename);

        Console.WriteLine("file downloaded");
    }
}
```

OpenRead -3 ووظيفتها إنشاء Read Only Stream بين الزبون والسيرفر لجلب بيانات من URL محدد وتخزينه في Stream Object بعد تمرير ال URL للموقع الذي تريد عرضه وباستخدام الميثود ReadLine نستطيع عرض البيانات المخزنة في ال Stream Object على شكل HTML Code .
ملاحظة : تستخدم الميثود Peek لمعرفة نهاية ال Stream Object .

```
using System;
using System.IO;
using System.Net;

class OpenRead_Method
{
    public static void Main ()
    {
        WebClient wc = new WebClient();
        string response;

        Stream strm = wc.OpenRead("http://www.google.com");
        StreamReader sr = new StreamReader(strm);

        while(sr.Peek() > -1)
        {
            response = sr.ReadLine();
            Console.WriteLine(response);
        }
        sr.Close();
    }
}
```

ويحتوي ال **WebClient Class** على مجموعة من ال **Properties** والتي تستخدم لجلب معلومات عن ال Web Host مثل ال ResponseHeaders property والذي يستخدم لجلب معلومات هامة عن ال web host مثل عدد ال Headers ونوع ال cash control واسم ال Server و نوع ال Encoding المستخدم وغيرها من المعلومات الهامة، ويستخدم كما يلي كمثال:

```
using System;
using System.Net;

class ResponseHeaders_property
{
    public static void Main ()
    {
        WebClient wc = new WebClient();
        byte[] response = wc.DownloadData("http://www.google.com");
        WebHeaderCollection whc = wc.ResponseHeaders;
        Console.WriteLine("header count = {0}", whc.Count);
        for (int i = 0; i < whc.Count; i++)
        {
            Console.WriteLine(whc.GetKey(i) + " = " + whc.Get(i));
        }
    }
}
```

```
//Output:
//header count = 6
//Cache-Control = private
//Content-Type = text/html
//Set-Cookie = PREF=ID=6ae22f44980c5d78...
//7JRA; expires=Sun, 17-Jan-2038 19:14:
//Server = GWS/2.1
//Transfer-Encoding = chunked
//Date = Wed, 23 Nov 2005 10:10:58 GMT
```

ثانيا : Uploading to Web Server

يدعم ال WebClient أربعة Methods لتحميل البيانات إلى ال Web Server وهي :
OpenWrite -1 يستخدم لإرسال Stream Data إلى ال Web Server وذلك بعد تمرير عنوان ال URL للملف والنص الذي نريد كتابته على ال Web Page طبعا يجب أن تملك الصلاحيات لذلك ويستخدم كما يلي كمثال :

```
using System;
using System.IO;
using System.Net;
class OpenWrite_method
{
    public static void Main ()
    {
        WebClient wc = new WebClient();
        string data = "<h1>Welcome to My Page</h1>";
        Stream strm = wc.OpenWrite("C:\\mypage.html");
        StreamWriter sw = new StreamWriter(strm);
        sw.WriteLine(data);
    }
}
```

```

        sw.Close();
        strm.Close();
    }
}

```

2 – UploadData ويستخدم لنقل محتويات مصفوفة من النوع Byte إلى ال Web Server وهذا يعني انك تستطيع من خلالها رفع أي نوع من البيانات مثل النص الصور الفيديو وغيره إلى ال web server بعد تحويلها إلى Byte Array ويستخدم كما يلي كمثال :

```

using System;
using System.Net;
using System.Text;

class UploadData_Method
{
    public static void Main ()
    {
        WebClient wc = new WebClient();
        string data = "This is The Text Before Converted it to Byte";
        byte[] dataarray = Encoding.ASCII.GetBytes(data);
        wc.UploadData("C:\\mydata.txt", dataarray);
    }
}

```

3- UploadFile وتستخدم هذه الميثود لرفع ملف من ال Local Computer إلى ال Web Host وهي بسيطة الاستخدام جدا وتستخدم كما يلي كمثال :

```

using System;
using System.Net;

class UploadFile_Method
{
    public static void Main ()
    {
        WebClient wc = new WebClient();
        wc.UploadFile("http://www.yoursite.com", "C:\\myfile.html");
    }
}

```

4- UploadValues وتستخدم لرفع **Collection** من البيانات وال values الخاصة بها إلى الويب سيرفر وذلك بعد تحويل ال **Collection** إلى Byte Array ولتعريف **Collection** نستخدم الكلاس NameValueCollection الموجود في اليم سبيس System.Collections.Specialized وبعد تعريفه نستخدم الميثود add لإضافة ال Collection جديد.. وتستخدم كما يلي كمثال :

```

using System;
using System.Collections.Specialized;
using System.Net;
using System.Text;

class UploadValues_Method
{

```

```

public static void Main ()
{
    WebClient wc = new WebClient();
    NameValueCollection nvc = new NameValueCollection();
    nvc.Add("firstname", "Fadi");
    nvc.Add("lastname", "Abdel-qader");
    byte[] response = wc.UploadValues("http://localhost/mypage.aspx", nvc);
    Console.WriteLine(Encoding.ASCII.GetString(response));
}
}

```

ثالثا:المواضيع الأكثر تقدما في ال HTTP Programming:

يعتبر هذا الجزء من أهم الأجزاء في برمجة تطبيقات Web Client Applications والذي سوف نتحدث فيه عن استخدام كل من ال HttpWebRequest Class و ال HttpWebResponse Class :

1- استخدام HttpWebRequest Class :

يحتوي هذا الكلاس على مجموعة من ال Properties والتي تستخدم بشكل أساسي في تطبيقات ال Web Client Applications لإنشاء مثل :

1- استخدام خاصية ال Web Proxy : والتي نمرر فيها عنوان ال Proxy Server ورقم البورت حتى نستطيع التعامل مع ال HTTP Web Requests من خلف Proxy Server أو Firewall ويتم تعريف ال Proxy Server Prosperity كما يلي كمثال :

```

using System;
using System.Net;

class ProxyServer_Property
{
    public static void Main ()
    {
        HttpWebRequest hwr = (HttpWebRequest)WebRequest.Create(
            "http://www.google.com");

        WebProxy proxysrv = new
        WebProxy("http://proxy1.server.net:8080");
        hwr.Proxy = proxysrv;

    }
}

```

نعرف في البداية ال HttpWebRequest Object ثم نعرف WebProxy Object من الكلاس webProxy ونسند له عنوان ال Proxy Server ورقم البورت وبعد ذلك نستطيع إسناده إلى أي اوبجكت باستخدام الخاصية Proxy التي تكون موجودة عادة في جميع .. HttpWebRequest Objects

2- استخدام ال HttpWebrequest لإرسال بيانات إلى الويب سيرفر باستخدام ال Streams وتستخدم كما يلي كمثال :

```

HttpWebRequest hwr =
    (HttpWebRequest)WebRequest.Create("http://localhost");
Stream strm = hwr.GetRequestStream();
StreamWriter sw = new StreamWriter(strm);
sw.WriteLine(data);

```

بعد تعريف ال HttpWebRequest Object نقوم بتعريف Stream Object ونسند له ال Request Stream من خلال الميثود GetRequestStream .

2 - استخدام HttpWebResponse Class:

تستخدم ال HttpWebResponse Object لإرجاع بيانات من الويب سيرفر إلى ال Client حيث نستخدم الميثود GetResponse و الميثود BeginGetResponse لهذه العملية ولا يوجد فرق في وظيفة هذه الميثودس سوى أن BeginGetResponse تعتبر asynchronous Method .

يحتوي ال HttpWebResponse Object على عدد من ال Properties وهي :

- 1- CharSet : وتستخدم لتحديد نوع ال Character Set
- 2- ContentEncoding : وتستخدم لعملية ال encoding
- 3- ContentLength : وتستخدم لمعرفة حجم الرد
- 4- ContentType : لتحديد نوع ال Response
- 5- Cookies : لتعامل مع ال Cookies ولستخدامها يجب أولاً إنشاء ملف Cookie فارغ وتعريفه كما يلي كمثال :

```

HttpWebRequest hwr =
    (HttpWebRequest)WebRequest.Create("http://www.amazon.com");
hwr.CookieContainer = new CookieContainer();

```

وذلك قبل ال HTTP Request ثم نسنده إليه كما يلي :

```

HttpWebResponse hwrsp = (HttpWebResponse)hwr.GetResponse();
hwrsp.Cookies = hwr.CookieContainer.GetCookies(hwr.RequestUri);

```

- 6- Headers : لمعرفة ال HTTP Headers
- 7- LastModified : يرجع فيه وقت وتاريخ آخر تعديل
- 8- Method : لمعرفة الميثود والتي تستخدم في ال HTTP Response
- 9- ProtocolVersion : لمعرفة ال HTTP Version
- 10- ResponseUri : ال URL الخاص بسيرفر
- 11- Server : لمعرفة اسم السيرفر
- 12- StatusCode : لمعرفة نوع ال Coding المستخدم
- 13- StatusDescription : لإرجاع ال Text يحتوي على حالة ال HTTP

في الدرس القادم سوف نتحدث عن ال Web Services & XML Programming إنشاء الله ..