

NETFX02 – HttpClient: richieste HTTP in tutte le salse

Carmine Punella

Development Engineer

carmine.punella@gmail.com

Twitter: [@cpunella](https://twitter.com/cpunella)

Dotnetlombardia.org/b/remixx/default.aspx

#CDays13 – 24, 25 e 26 settembre 2013, Roma

Grazie a

Sponsor



COMMUNITY DAYS 

Agenda

- HTTPClient
- PCL: codice comune!
- Portable HTTP Client

HTTP Client

- Cos'è e perchè?
- Versione attuale?
- Nuovo namespace: `System.Net.Http`;

Requisiti

- Visual Studio 2012
- .NET 4.5 / WinRT

Request!

1. Usare metodi **Get/Post/Put/Delete + Async**
2. Creare un **HttpRequestMessage** e passarlo al metodo **SendAsync**

Request!

DEMO

HttpWebRequest VS WebClient VS HttpClient

FIGHT!

HttpWebRequest VS HttpClient

- La prima è sicuramente la più completa ma...
 - Codice lungo da scrivere
 - Più difficile da comprendere
 - Troppe proprietà

Post di un file?

```
public static void HttpUploadFile(string url, string file, string paramName, string contentType, NameValueCollection nvc) {  
    string boundary = "-----" + DateTime.Now.Ticks.ToString("x");  
    byte[] boundarybytes = System.Text.Encoding.ASCII.GetBytes("\r\n--" + boundary + "\r\n");  
  
    HttpRequest wr = (HttpRequest)WebRequest.Create(url);  
    wr.ContentType = "multipart/form-data; boundary=" + boundary;  
    wr.Method = "POST";  
    wr.KeepAlive = true;  
    wr.Credentials = System.Net.CredentialCache.DefaultCredentials;  
  
    Stream rs = wr.GetRequestStream();  
  
    string formdataTemplate = "Content-Disposition: form-data; name=\"{0}\"{1}\r\n\r\n{2}";  
    foreach (string key in nvc.Keys)  
    {  
        rs.Write(boundarybytes, 0, boundarybytes.Length);  
        string formitem = string.Format(formdataTemplate, key, nvc[key]);  
        byte[] formitembytes = System.Text.Encoding.UTF8.GetBytes(formitem);  
        rs.Write(formitembytes, 0, formitembytes.Length);  
    }  
    rs.Write(boundarybytes, 0, boundarybytes.Length);  
  
    string headerTemplate = "Content-Disposition: form-data; name=\"{0}\"; filename=\"{1}\"{2}\r\n\r\nContent-Type: {3}\r\n\r\n";  
    string header = string.Format(headerTemplate, paramName, file, contentType);  
    byte[] headerbytes = System.Text.Encoding.UTF8.GetBytes(header);  
    rs.Write(headerbytes, 0, headerbytes.Length);  
  
    FileStream fileStream = new FileStream(file, FileMode.Open, FileAccess.Read);  
    byte[] buffer = new byte[4096];  
    int bytesRead = 0;  
    while ((bytesRead = fileStream.Read(buffer, 0, buffer.Length)) != 0)  
    {  
        rs.Write(buffer, 0, bytesRead);  
    }  
    fileStream.Close();  
  
    byte[] trailer = System.Text.Encoding.ASCII.GetBytes("--" + boundary + "--\r\n");  
    rs.Write(trailer, 0, trailer.Length);  
    rs.Close();  
  
    WebResponse wrsp = null;  
    try {  
        wrsp = wr.GetResponse();  
        Stream str = wrsp.GetResponseStream();  
        StreamReader sr = new StreamReader(str);  
    } catch (Exception ex) {  
        if (wrsp != null) wrsp.Close();  
    }  
    finally {  
        wr = null;  
    }  
}
```

45

```
_httpClient = new HttpClient();  
  
_httpClient.BaseAddress = new Uri("http://localhost",  
UriKind.Absolute);  
_requestUri = new Uri("/api/values", UriKind.Relative);  
  
var fi = new FileInfo(@"c:\dev\test.xml");  
  
var form = new MultipartFormDataContent();  
    form.Add(new StreamContent(fi.OpenRead()),  
    "\"file\"", "\"\" + fi.Name + "\"");  
  
await _httpClient.PostAsync(_requestUri, form);
```

12

WebClient VS HttpClient

	WebClient	HttpClient
Supporto WinRT	-	X
Progress Download	X	-
Richieste concorrenti	-	X
Supporto FTP	X	-
Testing	\	X
Async	X	X
AutoRedirect (302)	-	X

UN PAIO DI DOMANDE

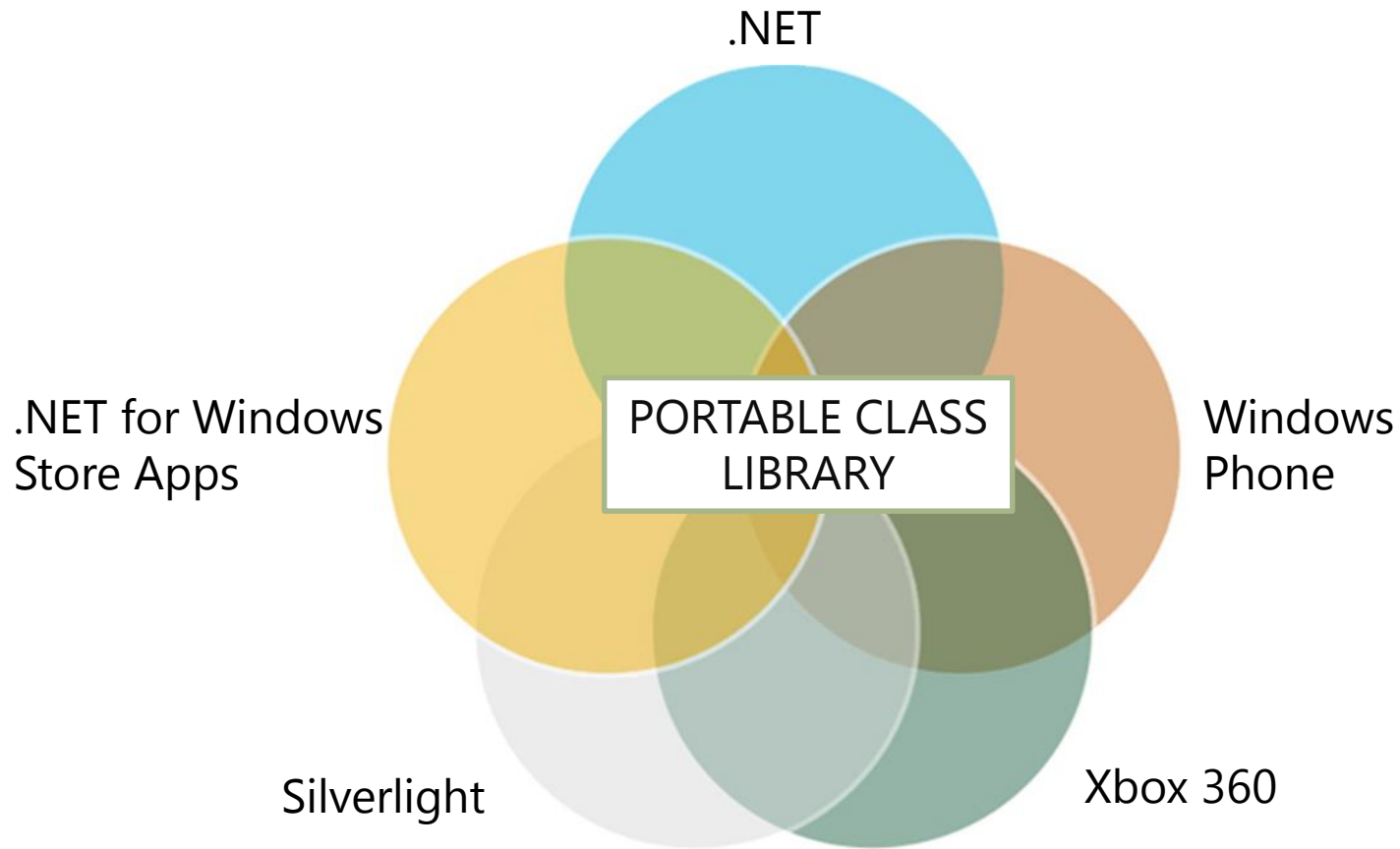
Quante piattaforme ci sono?

- Windows 8
- Windows Phone
- WPF
- Silverlight
- Xbox 360

Quante soluzione di creare codice
comune (fino a qualche mese fa)?

Zero

Microsoft... ci vuole bene! 😊



Definizione MSDN

Use this project to **write portable assemblies** that work **without modification** on multiple platforms, such as Windows 7, Windows 8, Silverlight, Windows Phone, and Xbox 360.

Vantaggi

- Scrivere della **business logic condivisa**
- **Evitiamo la duplicazione** del codice
(cosa sempre buona e giusta)
- **Tempi di sviluppo** più rapidi
- **Testing facilitato**
- Maggiore manutenibilità

Requisiti

- Versione \geq 2010 SP1
- Nella 2010 bisogna installare i [Portable Library Tools](#)
- Nella 2012 (in poi) supporto nativo

Build machine senza VS?

- Scaricare i [Portable Library Tools](#)
- Lanciare da riga di comando l'installazione aggiungendo come parametro **/buildmachine**

Features

Feature	.NET Framework	Windows Store	Silverlight	Windows Phone	Xbox 360
Core	√	√	√	√	√
LINQ	√	√	√	√	
IQueryable	√	√	√	7.5 and higher	
Dynamic keyword	Only 4.5	√	√		
Managed Extensibility Framework (MEF)	√	√	√		
Network Class Library (NCL)	√	√	√	√	
Serialization	√	√	√	√	
Windows Communication Foundation (WCF)	√	√	√	√	
Model-View-View Model (MVVM)	Only 4.5	√	√	√	
Data annotations	Only 4.0.3 and 4.5	√	√		
XLINQ	Only 4.0.3 and 4.5	√	√	√	√
System.Numerics	√	√	√		

Titolo demo

DEMO

Portable Library Releases

- [Portable HttpClient](#)
- [Microsoft.Bcl.Compression](#) (Zip, Gzip compression)
- [Microsoft.Bcl.Async](#) (Await/Async)
- Azure Mobile Services
- MVVM Light
- JSON .NET
- Ecc...

Ritornando al discorso che Microsoft... ci vuole bene! 😊

Portable Class Libray

HTTP Client

Requisiti minimi

- VS 2010
- .NET 4.0 / WP 7.5
- NuGet \geq 2.1

TIP: con .NET 4.0/WP 7.5 bisogna referenziare anche la libreria Microsoft.Bcl.Async

Portable HTTP Client in tutte le salse!

DEMO

Q&A

Tutto il materiale di questa sessione su
<http://www.communitydays.it/>

#CDays13