

# WEB02 – Le novità di ASP.NET 4.5

**Daniele Bochicchio**

**daniele@aspitalia.com**

**Twitter: @dbochicchio**

**blogs.aspitalia.com/daniele**

**Senior Sw Architect@5DLabs.it**

**#CDays12BA – 8 e 9 maggio**

 **aspitalia.com**

**5D** *labs.it*

# Grazie a

5Dlabs.it

Microsoft  
Visual Studio  
with MSDN Subscription

Powered by  
Windows Azure

Sponsor

Windows Phone  
Put people first.

edilportale

**HOEPLI**  
INFORMATICA

aspitalia.com

DotNetSide

UGIdotNET

# Agenda

- ASP.NET Core
  - Async requests/responses
  - Await e task-based modules e handlers
  - Lazy request validation e Anti-XSS Lib
  - WebSockets
  - Bundling e minification
- ASP.NET WebForms
  - Data control Strongly typed
  - Model binding
  - Validazione dell'input unobtrusive
  - HTML5
- Visual Studio 11
  - Smart tasks, HTML5, extract to user control, tag renaming, JavaScript, CSS
  - Page inspectors

# Quando esce ASP.NET 4.5?

- Nel 2012
- Ora è in preview con Windows 8/VS 11
  - <http://dev.windows.com>
- Refresh con Windows 8 Consumer Preview?

# Async requests

- ASP.NET 4.5 introduce il supporto a stream asincroni
  - `HttpRequest.GetBufferlessInputStream`
  - Questo metodo consente l'accesso async allo stream, ASP.NET rilascia il thread quando non è necessario
  - Lo stream restituito ha i metodi `BeginRead` e `EndRead`
- È compatibile con handler e modules esistenti
  - ASP.NET copia i dati
  - `GetBufferedInputStream` per accesso al buffer

# Async responses

- ASP.NET fa il buffer ed invia la risposta quando è pronta
- Questo può essere un problema con connessioni a bassa banda
- Si può disabilitare e chiamare ogni tot `HttpResponse.Flush`
- ASP.NET introduce i metodi `BeginFlush` e `EndFlush` per inviare dati in maniera async, senza bloccare thread
- Si applica sia a modules che handlers

# Await e task-based modules e handlers

- Fa uso delle novità di C# 5/VB 11 e della Task Parallel Library introdotta con v4
- Consente di semplificare di gran lunga le operazioni task-based
- Si applica a modules e handlers
- Non blocca i thread
  - Scalabilità, scalabilità, scalabilità! ;)

# HTTPModule async

- Nuova classe helper *EventHandlerAsyncHelper*, che semplifica la sottoscrizione di eventi
- Si fa un normale `HttpModule` di ASP.NET

# HTTPHandler async

- Classe *HttpTaskAsyncHandler* da usare come base, invece di implementare l'interfaccia *IHttpHandler*
- Noi dobbiamo solo fare l'override del metodo *ProcessRequestAsync*
- Il resto lo fa tutto ASP.NET :)

Async Handlers/Modules

# DEMO 01

# Lazy request validation

- Come ben saprete, ASP.NET verifica header, forms, querystring e cookie ad ogni richiesta
- In caso venga trovato un contenuto potenzialmente pericoloso, la chiamata va in errore
- ASP.NET 4.5 introduce la lazy request validation

# Lazy request validation

- Si attiva con il web.config
  - `<httpRuntime requestValidationMode="4.5" />`
- In questo modo ASP.NET controlla i dati solo quando effettivamente vi accedete
  - E solo per quelli a cui avete accesso
    - Es: `Request.Form["ID"]` non farà scattare i controlli sul resto dei valori inviati
- Si possono richiedere dati non validati
  - `context.Request.Unvalidated.Form["ID"];`
- *Da usare con attenzione!*

# AntiXSS Lib integrata

- La libreria AntiXSS ora è integrata nativamente
- Non è attiva di default per compatibilità

```
<httpRuntime  
    encoderType="System.Web.Security.AntiXss.AntiXssEncoder" />
```

- *HtmlEncode, HtmlFormUrlEncode, HtmlAttributeEncode, CssEncode, UrlEncode, UrlPathEncode, XmlEncode, XmlAttributeEncode*

Lazy request validation e AntiXSS Lib

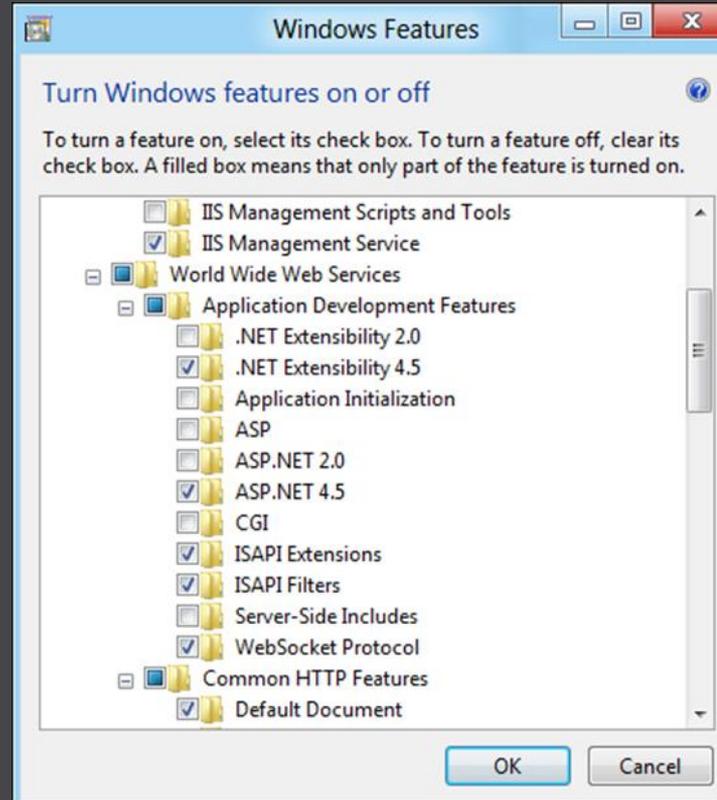
# DEMO 02

# WebSockets

- Parte delle specifiche di HTML5
- Socket bi-directional, full duplex
- Consente al server di mandare dati al client
- È il client che chiede l'upgrade da `http://` a `ws://`
- Nuovo namespace `System.Web.WebSockets`
- Supportato da IE10+, FF e Chrome v. *WFT+*

# Prerequisiti

- Va abilitato in Windows 8
- Va usato IIS
- Bisognare usare un browser che li supporti



WebSockets

**DEMO 03**

# Bundling e minification

- Bundling
  - Combino più file insieme
- Minification
  - " Compatto" il contenuto di un file
- In genere si applica a CSS e JavaScript
- Riduce latenza e consumo di banda
  - Meno richieste a file esterni, più veloce è la risposta del browser

Bundling e minification

**DEMO 04**

# Data control strongly typed

- Stanchi di `<%#Eval("ID")%>`?

```
<asp:Repeater ModelType="Customer"...>
```

```
  <ItemTemplate>
```

```
    <%#Item.ID%>
```

```
  </ItemTemplate>
```

```
</asp:Repeater>
```

- Nel two-way databinding si usa `BindItem` anziché `Item`

# Model binding

- Avete presente quello che fanno le ruote in MVC?
- Ora lo si può fare in WebForms :)
- In realtà incorpora concetti dell'ObjectDataSource e di ASP.NET MVC
- Niente più chiamate esplicite di `DataBind()`

# Model binding

- Lavora con IQueryable<T>
  - I controlli filtrano usando il provider LINQ corrispondente
    - Ordinamenti, paginazioni in nativo!
    - Se si usa Entity Framework, viene generata una query ad hoc per il db

Data control e model binding

# DEMO 05

# Unobtrusive validation

- Cosa c'è di più antico in ASP.NET? Il JavaScript dei validator control!
- ASP.NET 4.5 introduce i validator unobtrusive
  - Stile ASP.NET MVC, meno JavaScript da scaricare
- Si attivano:
  - Globalmente
    - Via `<appSettings>` in `web.config`
      - `<add name="ValidationSettings:UnobtrusiveValidationMode" value="WebForms" />`
    - Con la proprietà `System.Web.UI.ValidationSettings.UnobtrusiveValidationMode`
  - Localmente per una pagina
    - Impostando la proprietà `Page.UnobtrusiveValidationMode` su `UnobtrusiveValidationMode.WebForms`.

Unobtrusive validation

**DEMO 06**

# HTML5

- L'UpdatePanel funziona
- I controlli nativi (es: video) hanno il modello server
- TextBox ha una proprietà TextMode per gestire i nuovi tipo di input di HTML5
- VS ha il template e lo schema di default su HTML5

# VS11: HTML e ASP.NET

- Smart task sui controlli/HTML
- Supporto a WAI-ARIA per l'accessibilità
- Nuovi snippet HTML5
- Extract to User Control
- Intellisense per code-nudgets negli attributi HTML (ie: databinding!)
- Renaming automatico del tag di chiusura
- Event-handler da codice (stile XAML)
- **IISExpress** il server di default

# VS11: JavaScript

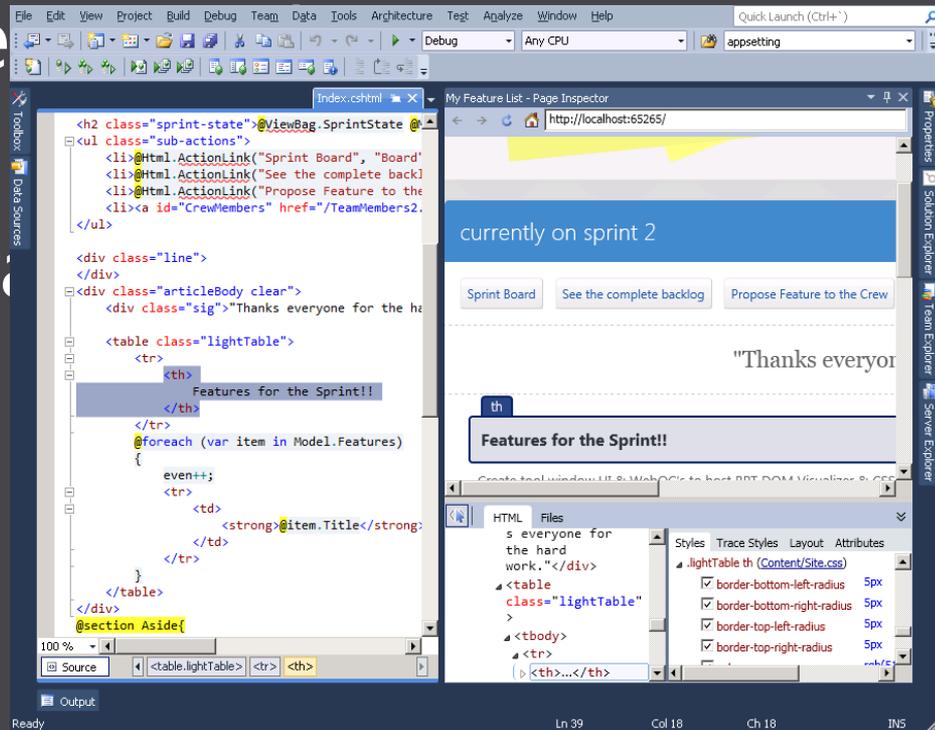
- Code outlining
- Matching delle {}
- Go to definition
- Supporto ad ECMAScript 5
- Intellisense sul DOM
- Migliorie a VSDOC

# VS11: CSS

- Auto-reduce completion per risultati più precisi
- Indentazione gerarchica
- Supporto ai CSS hack
- Supporto ai CSS vendor prefix
- Commenti veloci con CTRL+K,C/U
- Color picker
- Snippets
- Regions

# Page Inspectors

- Va installato a parte
- Consente di vedere codice HTML generato dal server.



VS11 e HTML5

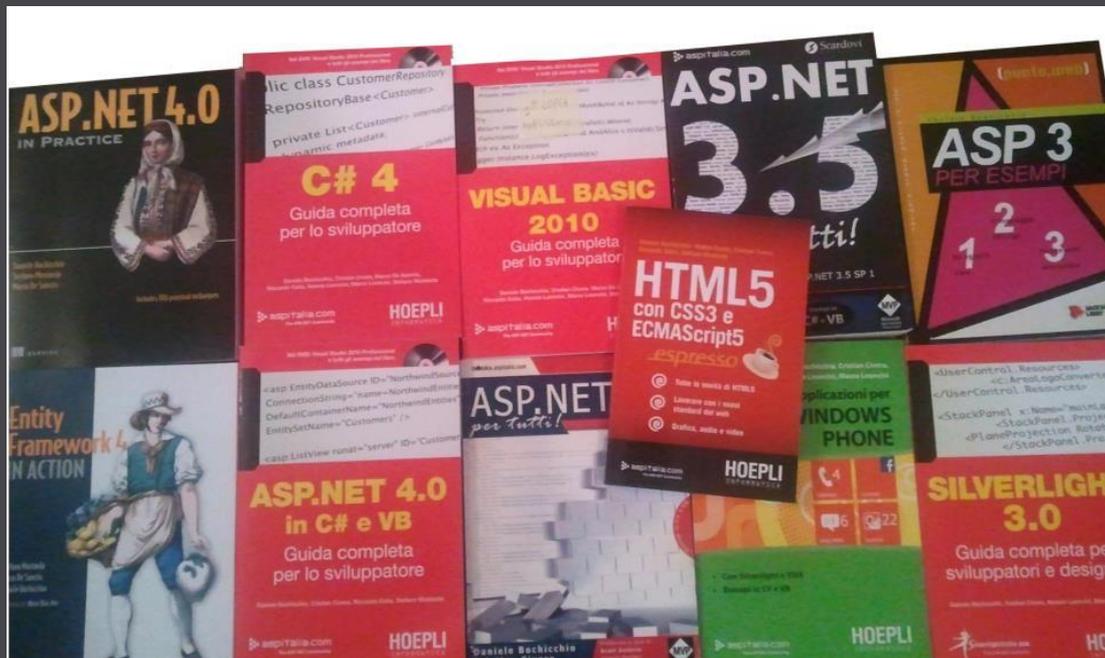
# DEMO 07

# Agenda recap

- ASP.NET Core
  - Async requests/responses
  - Await e task-based modules e handlers
  - Lazy request validation e Anti-XSS Lib
  - WebSockets
  - Bundling e minification
- ASP.NET WebForms
  - Data control Strongly typed
  - Model binder
  - Validazione dell'input unobtrusive
  - HTML5
- Visual Studio 11
  - Smart tasks HTML 5 extract to user control tag renaming

# Libri, libri, libri

- <http://books.aspitalia.com>



# Q&A

- Materiale su

<http://www.communitydays.it/>