

A LAP AROUND WEB APPLICATIONS WITH .NET 4.0

ASP.NET, MVC, Ajax

Ugo Lattanzi

*Software Architect @ Gaia
Microsoft MVP, MCP*

Twitter: @imperugo

Blog: <http://tostring.it>

E-mail: imperugo@gmail.com

Agenda

- ASP.NET 4.0
- MVC 2;
- Ajax

ASP.NET 4.0...

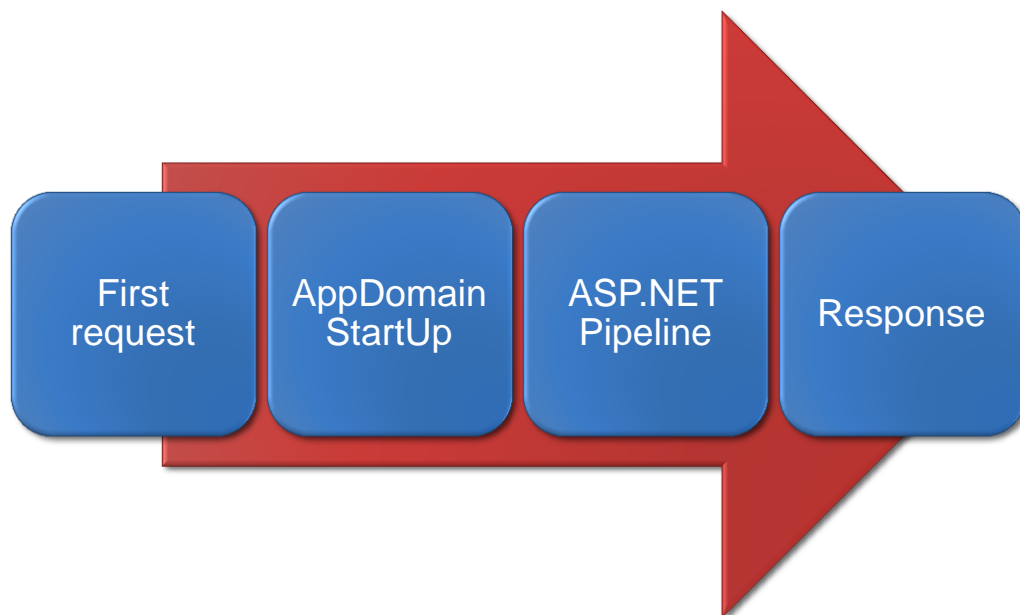
What's new

AUTOSTART

Partiamo dall'inizio, la prima richiesta 😊

Pronti, partenza, via...

- Alla prima richiesta le applicazioni web sono lente;
- Spesso necessitano di eseguire parecchio codice prima di poter evadere la prima richiesta;



...soluzione

- Specificare ad IIS 7.5 i siti che devono avviarsi automaticamente (**applicationHost.config**)

```
<sites>
→ <site name="MySite" id="1">
→ → <application path="/"
→ → → preloadEnabled="myPreloadImplementation"
→ </site>
</sites>
```

```
<preloadProviders>
→ <add name="myPreloadImplementation"
→ → type="imperugo.application.myPreloadClass,imperugo.application" />
→
</preloadProviders>
```

(myPreloadClass deve implementare **IProcessHostPreloadClient** presente in **System.Web.Hosting**)

SEO

Search Engine Optimization

Gestione SEO?

- Già con la Service Pack 1 del Framework 3.5 SP1 abbiamo la possibilità di creare con facilità url **SEO Friendly....**
- **...ora possiamo gestire facilmente il Redirect Permanent (301), Page Title, Description keyword;**



...così

- `Response.RedirectPermanent(«http://www.tostring.it»);`
- `Page.Title = "Thank's ASP.NET 4.0 for the SEO Optimization";`
- `Page.MetaKeywords = "ASP.NET 4.0, meta, SEO, keywords";`
- `Page.MetaDescription = «ASP.NET 4.0 Description»;`

SICUREZZA

<%: AutoEncode %>

Cross Site Scripting

- L'output HTML deve essere sempre Encodato!
- ERRORE:
 - `<% = myVar %>`
- CORRETTO
 - `<% = HttpUtility.Html(myVar) %>`
- OPPURE
 - `<%: myVar %>`



È estendibile

- Usiamo l'Anti Cross Site Scripting Library;

```
public class AutoEncodeWithAntiXss : HttpEncoder
{
    protected override void HtmlEncode(string value, TextWriter output)
    {
        output.Write(AntiXss.HtmlEncode(value));
    }

    protected override void HtmlAttributeEncode(string value, TextWriter output)
    {
        output.Write(AntiXss.HtmlAttributeEncode(value));
    }
}
```

```
<system.web>
  <httpRuntime
encoderType="Imperugo.Web.Test.AutoEncode.AntiXSS.AutoEncodeWithAntiXss,
Imperugo.Web.Test.AutoEncode.AntiXSS"/>
```

CACHING

Caching

- È importantissima, può essere il nostro paracadute!
- Finalmente è possibile specificare un provider custom, anche per l'outputCache;



```
<キャッシング>  
  <outputCache defaultProvider=" SossProvider ">  
    <providers>  
      <add name="SossProvider" type="imperugo.cache.SossOutputCacheProvider, imperugo.cache"/>  
      <add name="NCacheProvider" type="imperugo.cache.NCacheOutputCacheProvider,  
imperugo.cache"/>  
      <add name="FakeProvider" type="imperugo.cache.FakeOutputCacheProvider, imperugo.cache"/>  
    </providers>  
  </outputCache>  
</キャッシング>
```

CLIENT MODE ID

È la vera chicca!

- La sempre più alta presenza di javascript ci «impone» di avere il maggior controllo possibile sul markup ;
- ASP.NET introduce l'attributo ClientIdMode che permette di specificare come generare gli ID dei controlli:
 - **AutoID**: comportamento classico;
 - **Static**: l'ID non viene modificato dal motore di ASP.NET;
 - **Predictable**: usato nel DataBound insieme all'attributo ClientIDRowSuffix
 - **Inherit**: eredita la modalità dal controllo padre;



DEMO

WEB.CONFIG

w00t!

XDT (Xml Data Trasformation)

- Finalmente possiamo gestire facilmente più file di configurazione;
- Si integra perfettamente con la procedura di deploy;

```
<appSettings>  
  <add key="MyValue"  
    value="PREPRODUZIONE"  
    xdt:Transform="Replace"  
    xdt:Locator="Match(key)"/>  
</appSettings>
```



<http://tostring.it/blog/post/gestire-pi-and-ugrave-files-di-configurazione-con-visual-studio-2010-e-xdt-xml-data-trasformation>

DEMO

DEPLOY

Publish

- WebDeploy;
- FTP;
- Local System;
- FPSE (Front Page Server Extension);
- WebPackage (setup che include la configurazione di IIS, Certificati, etc);



DEMO

ALTRE FUNZIONALITÀ

Visual Studio 2010 Rulez

- Intellisense migliorata sul javascript;
- Numerosi nuovi snippets, tutto XHTML Compliant;
- Il Designer rispetta maggiormente l'HTML che viene generato;
- Il Designer genera CSS 2.1;

Controlli, Providers, etc

- Ora i controlli renderizzano XHTML Strict;
- Controllo del ViewState potenziato;
- Il SessionState permette di comprimere i dati (maggiori

per

```
<sessionState  
  mode="SqlServer"  
  sqlConnectionString="data source=mydbserver;Initial Catalog=myDatabase"  
  allowCustomSqlDatabase="true"  
  compressionEnabled="true"/>
```

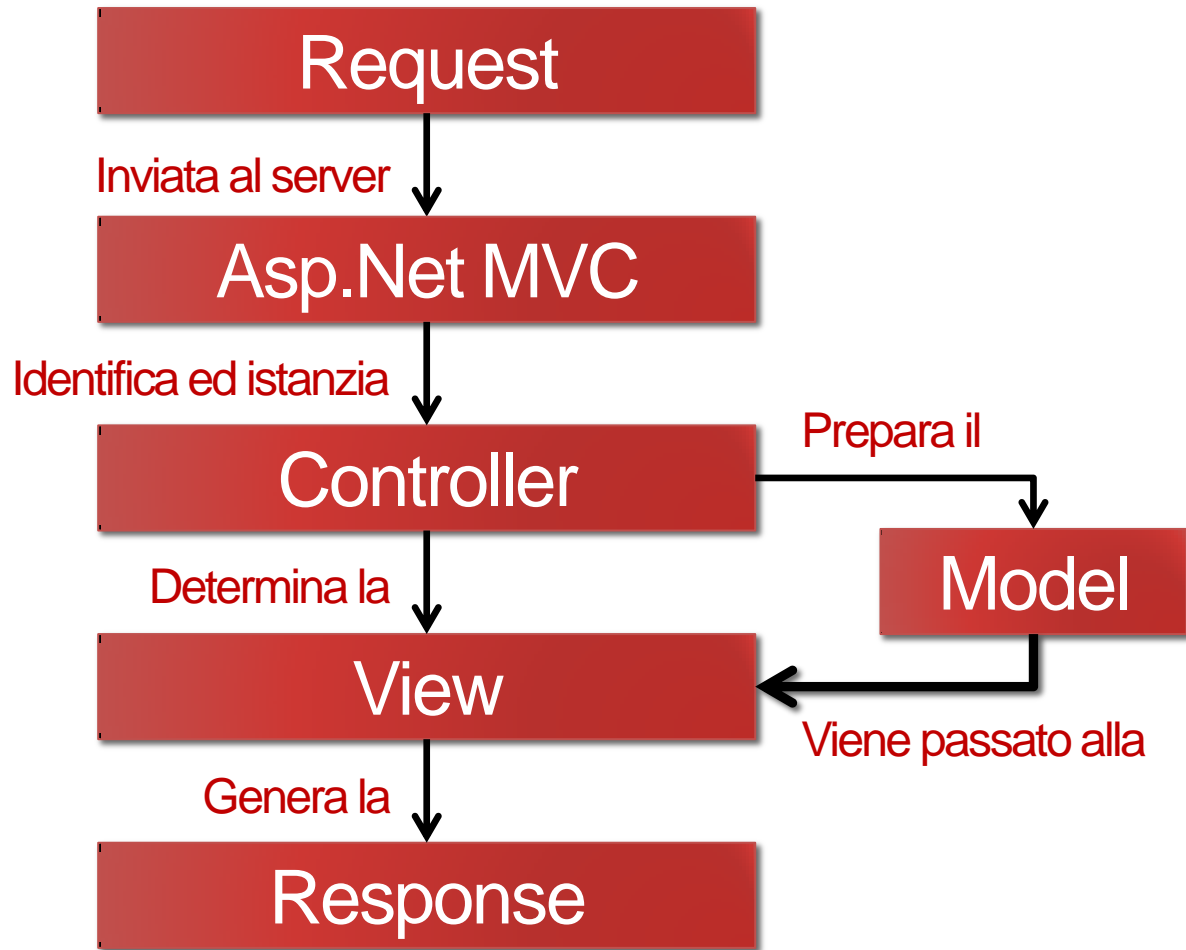
ASP.NET MVC2

Spettacolo!

Dove ci eravamo lasciati

- MVC permette di avere pieno controllo sull'html generato
- La logica di interfaccia è «unit-testabile»
- Open source
- Possiamo usarlo insieme alle WebForm

Come funziona?



MVC 2 è un continuo...

- ...non una rivoluzione;
- Incluso in VS 2k10;
- Upgrade facile;
- Controllo totale del markup;
- Codice testabile;

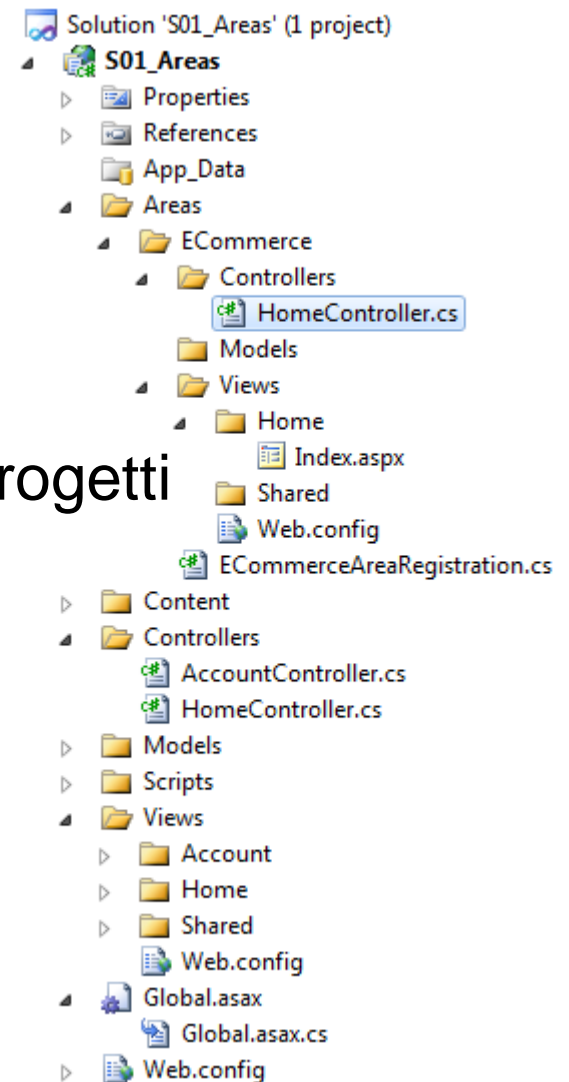
Produttività prima di tutto

- Areas;
- DataAnnotations;
- Templated Helpers;
- Validation;

AREAS

AREAS

- Un modo per dividere grossi progetti
- Permette uno sviluppo isolato per area
- Aiuta contro i conflitti di nomenclatura
- Configurazione singola o multipla dei progetti



DATA ANNOTATIONS

Data Annotations

- Viste per la prima volta in ASP.NET DynamicData
- Nuovi *attributi* che definiscono le regole di validazione o la logica di visualizzazione
- Sample annotations:
 - UIHint / DataType – Per override l'associazione del dato da visualizzare
 - DisplayFormat – Applica una stringa quando visualizziamo i dati
 - Required – Regola di validazione per rendere un campo obbligatorio

DEMO

TEMPLATED HELPERS

Templated Helpers

- Estensioni strongly-typed per HtmlHelper
- Associazione tra il tipo di dato ed il template visualizzato.
- Nuovi metodi di visualizzazione:
 - Labels (HtmlHelper.LabelFor)
 - Values (HtmlHelper.DisplayFor)
 - Editing (HtmlHelper.EditorFor)
- Ulteriore separazione tra il codice e la visualizzazione dei dati

Templated Helpers

- Views parziali (*.ascx)
- Aggiunto il supporto nelle strutture delle folders
- Associazione delle view parziali ai data type
- L'associazione può essere sovrascritta tramite le DataAnnotations o i metodi helper

DEMO

RENDER ACTION

Render Action

- `Html.RenderAction` permette di renderizzare una action da una vista “contenitore”
- Si può in questo modo aumentare la componentizzazione
- Grazie all'attributo `ChildActionOnly` è possibile specificare che un' azione è utilizzabile solo tramite `RenderAction`.

Sicurezza:

- AutoEncode supportato anche per MVC2 (FW4.0);
- ValidateAntiForgeryToken (potenziato);
- RequireHttps;
- Json and Get Request (retrocompatibile);

ASP.NET AJAX

Il mistero

È in continua evoluzione

- Open Source <http://aspnet.codeplex.com>
- È attualmente in beta;
- Ajax Control Toolkit embeddato nella libreria Microsoft AJAX principale;
- CDN;

Client Script Loader

- Si riferenzia solo lo script Microsoft Ajax (script.js);
- Grazie a Sys.Require() è possibile caricare i soli script necessari all'esecuzione della pagina:
 - Caricamento parallelizzato;
 - Evita il caricamento di script duplicati;
 - Gestione automatica delle dipendenze;
 - Utilizzabile anche per caricare script custom;
 - Intellisense;

Lo ScriptManager esiste ancora

- Il controllo ScriptManager ha due nuovi interessanti attributi
 - EnableCdn – permette di scegliere se gli script vengono caricati dalla cdn. Utilizzabile anche con script propri Es. [assembly: WebResource("Foo.js", "application/x-javascript", CdnPath = "http://foo.com/foo/bar/foo.js")]
 - AjaxFrameworkMode:
 - Disabled: non carica nessuno script base
 - Explicit: possibilità di scegliere cosa caricare
- CompositeScript: combina più script in uno solo ed automaticamente effettua encoding gzip

Q & A

GRAZIE MILLE

Ugo Lattanzi

*Software Architect @ Gaia
Microsoft MVP, MCP*

Twitter: @imperugo

Blog: <http://tostring.it>

E-mail: imperugo@gmail.com