{frameworks}

Fall in love with **ASP.NET MVC**

Testable Dynamic Web Applications with .NET

Mario Szpuszta

Senior Architect

Enterprise Services

Microsoft Austria

marioszp@microsoft.com

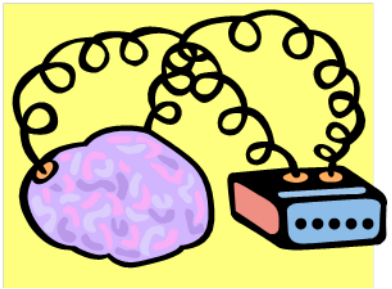
http://blogs.msdn.com/mszcool

October 5th, 2010

Denmark, Aarhus

What is ASP.NET MVC?

- Implementation of MVC for ASP.NET
- *Not a big deal, is it?*
  - *Simple, yet powerful*
  - *allows to focus*



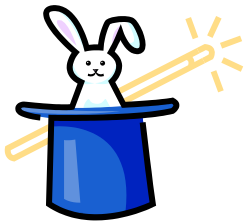
The 3 Key Questions…

How does it work, what does it mean?

How does it relate to other ASP.NET parts?

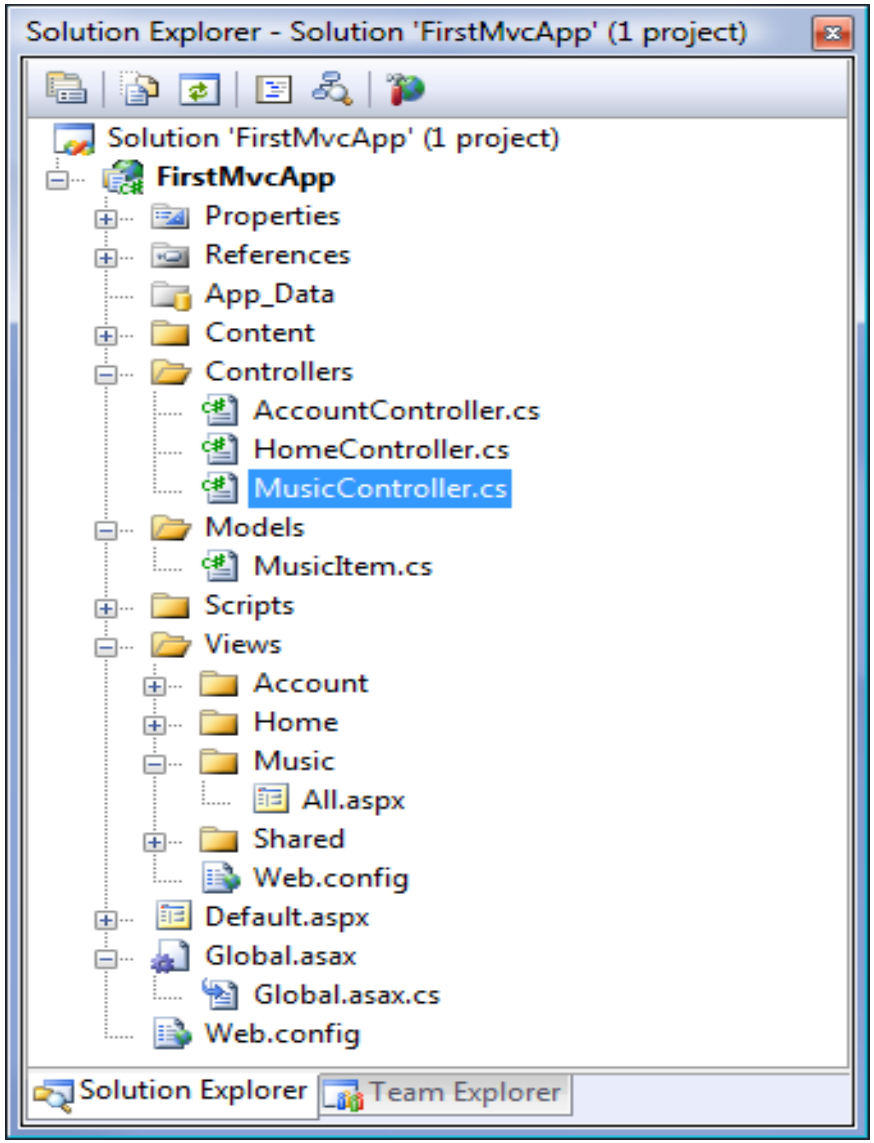
Why should we care? What's it's benefits?

MVC: What it means for us…



Default Behavior of ASP.NET MVC

- **Convention over configuration**
  - Special folders
  - Naming of items
- Less error-prone



So what does it mean?

Think in controllers & actions

(not pages and links)

Controllers, Actions and AJAX?

- You have full control on HTML
  - Can use any pure JavaScript-library you want
  - ASP.NET MVC documentation prefers jquery
- Server-side AJAX – very “light-weight”
  - Calling controller actions & processing results

```
<script src="/Scripts/MicrosoftAjax.js"
<script src="/Scripts/MicrosoftMvcAjax.js"
<script src="/Scripts/jquery-1.4.1.min.js"
```

```
<%: Ajax.ActionLink("Wake-up all machines of group",
                    "WakeUpGroup",
                    "MachineWakeup",
                    new RouteValueDictionary()
                    {
                        { "id", Model.Group.GroupId }
                    },
                    new AjaxOptions()
                    {
                        OnSuccess = "OnWakeUpSuccess",
                        OnFailure = "OnWakeUpFailed"
                    })%>
```

So what does it mean?

Think in HTTP-verbs and URIs.

(not just request and response)

So what does it mean?

Clearly separate concerns.

Keep things simple and small!

The 3 Key Questions…

How does it work, what does it mean?

How does it relate to other ASP.NET parts?

Why should we care? What's it's benefits?

ASP.NET – Core Architecture

- All based on existing foundation

Using ASP.NET Foundations

- You can re-use anything
  - As long as it is not „handler-specific“

- HttpModules, Request/Response etc.
- Cache, Session, Application
- Master pages, content pages
- Security, Forms/Memberhsip/Roles
- Profiles, site maps, localization

The 3 Key Questions…

How does it work, what does it mean?

How does it relate to other ASP.NET parts?

Why should we care? What's it's benefits?

The key-benefits are:

- Frictionless Testability
- Tight control over <markup>
- User/SEO friendly URLs
- Leverage the benefits of ASP.NET
- Conventions and Guidance

MVC Workings – Clean URLs

- REST-like
  - /products/update
  - /blog/posts/2008/08/12/mvc-is-cool
- Friendlier to humans
  - Hack /product.aspx?categoryid=123 to become
  - more beautiful /products/kittens/
- Friendlier to web crawlers
  - Search engine optimization (SEO)

Testability and ASP.NET MVC

- Test controllers directly in unit-tests
  - It's just functionality
  - Controller never writes response directly
  - All results of a controller are *intentions*
- Key-requirement: separation of concerns
  - ASP.NET MVC allows SoC, but *we* write the code
- Mock if e.g. accessing session & co
  - Rhino - http://ayende.com/projects/rhino-mocks/downloads.aspx
  - Moq library - http://code.google.com/p/moq/

The 3 Key Questions…

How does it work, what does it mean?

How does it relate to other ASP.NET parts?

Why should we care? What's it's benefits?

The 3 Key Answers…

Think in controllers, actions and HTTP-verbs!

Built on of ASP.NET foundation, so leverage it!

Separation of concerns, nice URLs, testing!

Resources and Links

- www.asp.net/mvc
- http://channel9.msdn.com/search?term=MVC&x=0&y=0

- http://mvcmusicstore.codeplex.com/
- http://nerddinner.codeplex.com/

- http://www.hanselman.com/blog
- http://www.hanselman.com/blog/ASPNETMVCPreview3.aspx

- http://blogs.msdn.com/mszcool

Thank You!

Mario Szpuszta
Senior Architect
Enterprise Services
Microsoft Austria

marioszp@microsoft.com
http://blogs.msdn.com/mszcool
!