ASP.NET Web Architecture Design - Part 2

Asked 15 years, 11 months ago Modified 15 years, 11 months ago

Viewed 374 times

▲

2

▼

I've actually asked about this already in this post although we've gone back to the drawing board and found it's much more difficult than we initially thought. We develop a large (and very much relied upon) Intranet system which has evolved tremendously over the years.

The problem is that it uses frames to have a menu on the left, a header at the top and the main page thereafter. This is done to make sure the other elements are static whilst you're able to scroll through the pages. Now we've come up with a solution to achieve this using a single page and CSS, but the problem lay in the actual architecture of our web application.

We're wanting to shift to MVC architecture to make our pages more secure and manageable, however there is a major problem with this. Each section of our Intranet system has cross-links to other pages in other sections. So, for example, say we had a section called "Games" and in that we had a section called "People Playing Games" but this was actually just using a frame to point to a page which is in another folder on our Intranet system called "Personnel" then "People.aspx" (a pure example). How would the MVC architecture deal with something like that?

We have pages displayed in completely different areas of our Intranet that are in actual fact the same page. So having a URL like http://mysite/section/category/page would have to link to http://mysite/anothersection/differentcategory/page.

Hopefully this makes sense to people with experience in MVC architecture as well as pre-dated architectures such as the use of frames like we're doing.

The questions raised would be:

- Do we have to have duplicate pages?
- Is there a way to cross-reference a page in different sections using MVC?
- For an Intranet system consisting of about 400 different pages, is MVC the way to go?

Cheers

asp.net-mvc

Share

Improve this question

Follow

edited May 23, 2017 at 10:32

Community Bot

1 ● 1

asked Jan 13, 2009 at 15:18

Kieran Senior
18.2k ● 27 ● 96 ● 139

4 Answers

Sorted by: Highest score (default) ⬍

2

- You shouldn't have to, just like the idea behind complex types. You can drop in min-views into a full fledged view to separate repetition.
- You can use Html.ActionLink for creating links to pages and you can use Html.RenderPartial or Html.RenderAction to render a whole action into your page.
- For sure. Like I said you can break up repetition into small components and include them when you want.

For example...

I am working on a small storefront application for a friend and I found myself creating a view for Adding and Editing an Product. Problem is -- I was using the same layout for the fields. So I created a MVC User Control and dropped in the fields and did something like this.

For Add..

```
<asp:Content ID="DefaultContent"
ContentPlaceHolderID="DefaultContentPlaceHolder" runat
<% Html.BeginForm<CatalogController>(c => c.AddProduct
%>
<% Html.RenderPartial("ProductFields", ViewData); %>
<%= Html.SubmitButton("Submit", "Add") %>
```

```
<% Html.EndForm(); %>
</asp:Content>
```

And for Edit...

```
<asp:Content ID="DefaultContent"
ContentPlaceHolderID="DefaultContentPlaceHolder" runat
<% Html.BeginForm<CatalogController>(c => c.EditProduc
FormMethod.Post); %>
<% Html.RenderPartial("ProductFields", ViewData); %>
<%= Html.Hidden("Product.ID", ViewData.Model.ID) %>
<%= Html.SubmitButton("Submit", "Save") %>
<% Html.EndForm(); %>
</asp:Content>
```

And the ProductFields partial looked like this

```
<% Product product = ViewData.Model; %>
<% IEnumerable<Category> categories = ViewData["Catego
IEnumerable<Category>; %>
<table>
    <tr>
        <td><%= Html.Label("Product.Name", "Name") %><
        <td><%= Html.TextBox("Product.Name", product.N
    </tr>
    <tr>
        <td><%= Html.Label("Product.Price", "Price") %
        <td><%= Html.TextBox("Product.Price", product.
    </tr>
    <tr>
        <td><%= Html.Label("Product.Description", "Des
        <td><%= Html.TextBox("Product.Description", pr
</td>
    </tr>
    <tr>
        <td><%= Html.Label("Product.IsActive", "Is act
        <td><%= Html.CheckBox("Product.IsActive", prod
    </tr>
    <tr>
        <td><%= Html.Label("Product.IsFeatured", "Is f
        <td><%= Html.CheckBox("Product.IsFeatured", pr
? product.IsFeatured.Value : false) %></td>
    </tr>
    <tr>
        <td><%= Html.Label("Product.CategoryID", "Cate
        <td><%= Html.DropDownList("Product.CategoryID"
SelectList(categories, "ID", "Name")) %></td>
    </tr>
</table>
```

So as you can see you can easily break up your pages as needed and include them and pass data into them very easily with ASP.NET MVC.

Share Improve this answer Follow

answered Jan 13, 2009 at 15:26

Chad Moran
12.9k ● 2 ● 52 ● 72

Unfortunately my MVC knowledge when it comes to ASP.NET MVC is quite limited, so understanding this example is difficult. Can I just have another page in another section using Html.ActionLink with the page as a parameter or something? – Kieran Senior Jan 13, 2009 at 16:00

Html.ActionLink generates a URL based on a Controller/Action. That way your URLs don't get stale if you change your routing structure. – Chad Moran Jan 13, 2009 at 16:26

1

Kezzer - are you sure that you want to take an existing (and working, I assume) system and completely rewrite it just for the purported benefits of MVC? I know that this doesn't exactly answer your question, but I would seriously consider whether an MVC architecture will truly make your customers happier than a Webforms architecture - especially since they'll never really notice.

In a somewhat related vein, take a look at Joel's article on Architecture Astronauts. It isn't specific to your issue but it does address the unrealistic expectations that we sometimes get when looking at new technologies.

Share Improve this answer Follow

answered Jan 13, 2009 at 15:25

Mark Brittingham
28.9k ● 12 ● 82 ● 111

I like this post a lot, and you make a very good point. The problem we're having is that the system is evolving so rapidly that it's becoming difficult to maintain, especially with the use of frames. We have separate index.htm pages for every single section. I can see your point fully, however.
– Kieran Senior Jan 13, 2009 at 15:50

0

I'll answer the third part of your question about MVC being the way to go:

MVC is much easier to work on as a team. It works especially well in a designer/programmer tandem because the designer can work on what he/she knows and the programmer can work on what he knows (let's face it, there are very few she programmers) and instead of working in the same space (the .aspx page) the designer can be working in the .aspx view space while the programmer can be working in the .cs controller space.

There is also the added advantage of using one page many times. Using several different ActionNames and with a little bit of configuration with your partial pages and ViewData variables, you can use one page over and over and over again to cover almost any use case. I've found it very hard to do a similar configuration with WebForms.

The routing controls that you have in MVC are incredibly robust and with smart controller and view design you can end up making a 400 page Intranet boil down to 20 or so controllers with 80 pages. I'm rebuilding an ESPN-like intranet site that was done on WebForms that had close to 200 pages, and it all fits nicely within 8 controllers and 50 views.

Share Improve this answer Follow

answered Jan 13, 2009 at 15:43

thaBadDawg
5,220 ● 6 ● 36 ● 44

0

- Do we have to have duplicate pages?

No.

- Is there a way to cross-reference a page in different sections using MVC?

Yes.

- For an Intranet system consisting of about 400 different pages, is MVC the way to go?

as theBadDawg mentioned, as long as you can categorize those 400 pages into groups of pages that have something in common this is doable. Based on your **categories** and **sections** it sounds like have a logical grouping of pages.

Share Improve this answer

Follow

edited May 23, 2017 at 10:27

Community Bot

1 ● 1

answered Jan 13, 2009 at 18:23

Todd Smith

17.3k ● 11 ● 62 ● 82

We do, but they're sparse, and even in the same category can be highly unrelated. It's an odd system I must admit, although I'm in a meeting this week to discuss its overhaul. Are there any decent resources (books) to learn these things then? – Kieran Senior Jan 13, 2009 at 21:39