Which way do you prefer to create your forms in MVC?

Asked 16 years, 3 months ago Modified 5 years, 10 months ago Viewed 318 times

4

Which way do you prefer to create your forms in MVC?

```
<% Html.Form() { %>
<% } %>
```

Or

```
<form action="<%= Url.Action("ManageImage", "UserAccount") %>" method="post">
</form>
```

I understand that Html.Form() as of PR5 now just uses the URL provided by the request. However something about that doesn't sit well with me, especially since I will be getting all the baggage of any querystrings that are included.

What is your take?

asp.net-mvc forms model-view-controller

Share
Improve this question
Follow

edited Jan 26, 2019 at 18:31
Javier Enciso
55 ● 1 ● 2 ● 11

asked Sep 3, 2008 at 18:41
Nick Berardi
54.8k ● 15 ● 117 ● 136

6 Answers

Sorted by: Highest score (default)

7

✓

The second way, definitely. The first way is programmer-centric, which is not what the V part of MVC is about. The second way is more designer centric, only binding to the model where it is necessary, leaving the HTML as natural as possible.

Share Improve this answer Follow

answered Sep 3, 2008 at 18:55
user1228

1 I find myself doing it the first way, mostly. I'm a hypocrite. – user1228 Jan 12, 2009 at 3:43

3

On the whole, I think I'm kinda old-school as I prefer to roll my own HTML elements.

I also prefer a view engine like like NHaml, which makes writing HTML almost an order of magnitude simpler.

Share Improve this answer Follow

answered Sep 3, 2008 at 18:46
Brad Wilson
70.4k ● 9 ● 77 ● 85

1

I have to agree with both of you, I am not really like this simplistic WebForms style that seems to be integrating its way in to MVC. This stuff almost seems like it should be a 3rd party library or at the very least an extensions library that can be included if needed or wanted.

Share Improve this answer Follow

answered Sep 3, 2008 at 20:27
Nick Berardi
54.8k ● 15 ● 117 ● 136

1

I am totally in the opinion of old school HTML, that is what designers use. I don't like to include to much code centric syntax for this reason. I treat the web form view engine like a third party library, because I replaced it with a different view engine. If you do not like the way the web form view model works or the direction it is going, you can always go a different route. That is one of the main reasons I love ASP.NET MVC.

Share
Improve this answer
Follow

edited May 23, 2017 at 12:19
Community Bot
1 ● 1

answered Sep 4, 2008 at 0:52
Dale Ragan
18.3k ● 3 ● 55 ● 71

1

I agree with Andrew Peters, DRY. It should also be pointed out that you can specify your controller, action, and params to the .Form() helper and if they fit into your routing rules then no query string parameters will be used.

I also understand what Will was saying about the V in MVC. In my opinion I do not think it is a problem to put code in the view **as long as it is for the view**. It is really easy to cross the line between controller and view if you are not careful. Personally I can not stand to use C# as a template engine without my eyes bleeding or getting the urge to murder someone. This helps me keep my logic separated, controller logic in C#, view logic in brail.

Share Improve this answer Follow

answered Sep 4, 2008 at 19:46

Andrew Burns
14.4k ● 9 ● 41 ● 42

0

The reason for using helpers is that they allow you to encapsulate common patterns in a consistent and DRY fashion. Think of them as a way of refactoring views to remove duplication just as you would with regular code.

For example, I blogged about some RESTful NHaml helpers that can build urls based on a model.

Share Improve this answer Follow

answered Sep 4, 2008 at 2:16

Andrew Peters
11.3k ● 4 ● 38 ● 34