

ASP.NET MVC ViewData (using indices)

Asked 16 years, 3 months ago Modified 8 years, 9 months ago Viewed 2k times



1



I had a working solution using ASP.NET MVC Preview 3 (was upgraded from a Preview 2 solution) that uses an untyped ViewMasterPage like so:

```
public partial class Home : ViewMasterPage
```

On Home.Master there is a display statement like this:

```
<%= ((GenericViewData)ViewData["Generic"]).Skin %>
```

However, a developer on the team just changed the assembly references to Preview 4.

Following this, the code will no longer populate ViewData with indexed values like the above.

Instead, ViewData["Generic"] is null.

As per [this question](#), ViewData.Eval("Generic") works, and ViewData.Model is also populated correctly.

However, the reason this solution isn't using typed pages etc. is because it is kind of a legacy solution. As such, it is impractical to go through this fairly large solution and update all .aspx pages (especially as the compiler doesn't detect this sort of stuff).

I have tried reverting the assemblies by removing the reference and then adding a reference to the Preview 3 assembly in the 'bin' folder of the project. This did not change anything. I have even tried reverting the Project file to an earlier version and that still did not seem to fix the problem.

I have other solutions using the same technique that continue to work.

Is there anything you can suggest as to why this has suddenly stopped working and how I might go about fixing it (any hint in the right direction would be appreciated)?

asp.net-mvc

edited May 23, 2017 at 11:45

asked Sep 15, 2008 at 2:25

Share

Improve this question

Follow



Community Bot

1 • 1



Matt Mitchell

41.8k • 35 • 121 • 185

5 Answers

Sorted by: Highest score (default)



1

We made that change because we wanted a bit of symmetry with the [] indexer. The Eval() method uses reflection and looks into the model to retrieve values. The indexer only looks at items directly added to the dictionary.



Share Improve this answer Follow

answered Sep 18, 2008 at 16:04



Haacked

59k • 14 • 91 • 115



0

I've decided to replace all instances of ViewData["blah"] with ViewData.Eval("blah"). However, I'd like to know the cause of this change if possible because:



1. If it happens on my other projects it'd be nice to be able to fix.
2. It would be nice to leave the deployed working code and not overwrite with these changes.
3. It would be nice to know that nothing else has changed that I haven't noticed.



Share Improve this answer Follow

answered Sep 15, 2008 at 2:45



Matt Mitchell

41.8k • 35 • 121 • 185



0

How are you setting the viewdata? This works for me:

Controller:

```
ViewData["CategoryName"] = a.Name;
```



View:

```
<%= ViewData["CategoryName"] %>
```



BTW, I am on Preview 5 now. But this has worked on 3 and 4...

Share Improve this answer Follow

answered Sep 15, 2008 at 3:45



Ricky

5,377 ● 2 ● 33 ● 30



Re: Ricky

0

I am just passing an object when I call the View() method from the Controller.



I've also noticed that on my deployed server where nothing has been updated, ViewData.Eval fails and ViewData["index"] works.



On my development server ViewData["index"] fails and ViewData.Eval works...



Share Improve this answer Follow

answered Sep 15, 2008 at 4:02



Matt Mitchell

41.8k ● 35 ● 121 ● 185



Yeah, so whatever you pass into the View is accessible in the View as ViewData.Model. But that will be just a good old object if you don't do the strongly typed Views...

0



Share Improve this answer Follow

answered Sep 15, 2008 at 14:16



Ricky

5,377 ● 2 ● 33 ● 30



Yeah, I get that. The problem was that ViewData["blah"] stopped getting set and I wanted to know why (guessing MVC Preview 4 changed this but wanted to confirm). I guess I was also wondering why I couldn't seem to undo the developer's reference changes – [Matt Mitchell](#)
Sep 15, 2008 at 23:28