

MVC Shared Partial Views using different controllers and models

Asked 6 years, 8 months ago Active 4 years, 7 months ago Viewed 15k times



I have 2 controllers that generate 2 index views. What i would like to do is use these views as global shared partial views but cant seem to get this working.

1

Does anyone know if this is even possible?



My controller code is



```
public ActionResult Index()
{
    var viewModel = (from P in db.Projects
                     join R in db.Reports on P.ProjectTitle equals
R.ReportProjectID into ps
                     from R in ps.DefaultIfEmpty()
                     select new MyViewModel { Project = P, Report = R });

    return View(viewModel);
}
```

My ViewModel code is

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MiLife2.ViewModels
{
    public class MyViewModel
    {
        public Project Project { get; set; }
        public Report Report { get; set; }
    }
}
```

and my view is

```

@model IQueryable<Milife2.ViewModels.MyViewModel>

@{
    ViewBag.Title = "Index";
}
enter code here

<h2>Index</h2>
<div>@Html.Partial("_Partial1")</div>
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table>
    <tr>
        <th></th>
    </tr>

    @foreach (var item in Model)
    {
        <tr>
            <td>@item.Project.ProjectTitle </td>
            <td>@item.Project.ProjectCreatedByID</td>
            <td>@item.Project.ProjectCreatedDate</td>

            <td>@if (item.Report == null)
            {
                <text>No Reports</text>
            }
            else
            {
                @item.Report.Title;
            }
            </td>
            <td>@if (item.Report == null)
            {
                <text> </text>
            }
            else
            {
                @item.Report.Description;
            }</td>

            <td>
                @Html.ActionLink("Edit", "Edit", new { id=item.Project.ProjectID }) |
                @Html.ActionLink("Details", "Details", new { id=item.Project.ProjectID }) |
                @Html.ActionLink("Delete", "Delete", new { id=item.Project.ProjectID })
            </td>
        </tr>
    }
</table>

```

```

    </tr>
  }
</table>

```

If i create a partial page and paste the above view into it and then use `@HTML.Partial("_ProjPartial")` i get the error

The model item passed into the dictionary is of type 'System.Collections.Generic.List 1[MiLife2.Project]', but this dictionary requires a model item of type 'System.Linq.IQueryable 1[MiLife2.ViewModels.MyViewModel]'.

This does not happen if i use `@HTML.Partial("_ProjPartial")` from within the Index cshtml page in the specific controller views folder.

c# asp.net-mvc asp.net-mvc-4 controller partial-views

asked Sep 25 '13 at 13:52



JMB

37 1 3 6

If you need to render an output of controller action to a view you should consider the [Html.RenderAction\(\)](#) method. – Jakub Sep 25 '13 at 13:54

Im new to MVC so Im not too sure how `HTML.RenderAction()` works. Should i render the controller action in the partial view and then use my index.cshtml coding? – JMB Sep 25 '13 at 13:58

Why do you want to pass an `IQueryable` to the View??? – margabit Sep 25 '13 at 14:02

@JMB Say you have a controller X with an actionmethod `X.Partial()` which is using a partial view `"_partial.cshtml"`. Then you have controller Y with an actionmethod `Y.Index()` which uses a view `"index.cshtml"` and in that index you want to render that partial view. So in that index you call `Html.RenderAction("X", "Partial")` and you should get it. – Jakub Sep 25 '13 at 14:06

2 Answers

Active

Oldest

Votes



From the error it looks like to me that your partial view is looking for the same model as you have on your view. Passing the model to your partial should fix that error

3



```
@Html.Partial("_Partial1", Model)
```

update:



since that didn't work for you I would try using an ajax call

```
$('.btnSubmit').on('click', function(){
    $.ajax({
        url: "@(Url.Action("Action", "Controller"))",
        type: "POST",
        cache: false,
        async: true,
        data: { id: id },
        success: function (result) {
            $(".Content").html(result);
        }
    });
});
```

then in your controller

```
public PartialViewResult GetPartial()
{
    var viewModel = (from P in db.Projects
                     join R in db.Reports on P.ProjectTitle equals R.ReportProjectID
    into ps
                     from R in ps.DefaultIfEmpty()
                     select new MyViewModel { Project = P, Report = R });

    return PartialView("_Partial1", viewModel);
}
```

Using this ajax call you can call the partial view from any view and you can pass different id's, on button clicks or as needed to refresh the view. Hopefully calling it this way will fix your error. let me know if you have any questions.

edited Sep 25 '13 at 16:26

answered Sep 25 '13 at 14:18



Matt Bodily

6,176 ● 4 ● 23 ● 43

This doesn't seem to have made any difference as i still get the same error.Does it have anything to do with wanting to have the partial view using a controller that is not part of the view folder. – [JMB](#) Sep 25 '13 at 14:38

when I have seen that error in the past it has just had to do with the model being passed to the partial. what does the @model line look like on your partial view? – [Matt Bodily](#) Sep 25 '13 at 15:14

'@model IQueryable<MiLife2.ViewModels.MyViewModel>' – JMB Sep 25 '13 at 15:16

so they do match and what I put in my answer should have fixed it. Is the error message still the same? Thinking you should try loading the partial with an ajax call to see if that works for you. will put the code in my answer in a min. – Matt Bodily Sep 25 '13 at 15:22

Yes the error is the same. Ill see if i can add the ajax call – JMB Sep 25 '13 at 15:35

Recently ran into something similar, so I wanted to add my 2 cents. The answer for me was in what I was passing to the Partial View.

I was attempting to pass a string to a partial view, but when that string happened to be `null`, it was acting as if I had not passed anything into the Partial, which means it defaulted to passing the the current view's model.

For example, I have a view which renders a partial and that partial takes in a string:

```
@model SomeModel

@{ Html.RenderPartial("_MyPartialView", SomeModel.StringProperty) }
```

If `SomeModel.StringProperty` happens to be `null`, then it is going to try and pass what ever the current view's model is (which, in this case is `SomeModel`). So instead, I simply wrote the following which will pass in an empty string if `SomeModel.StringProperty` happens to be null:

```
@model SomeModel

@{ Html.RenderPartial("_MyPartialView", SomeModel.StringProperty ?? string.Empty) }
```

Hope this helps someone.

answered Nov 16 '15 at 17:15



[hvaughan3](#)

8,854 ● 4 ● 39 ● 65