How to Return partial view of another controller by controller?

Asked 6 years, 5 months ago   Active 3 years, 4 months ago   Viewed 162k times

▲

76

▼

🔖

26

↺

I have an `XXX.cshtml` file in a `Views\ABC` folder. Its controller is `ABC`

I also have an action method in my `DEF` controller that return a `Partialview("XXX" , xyzmodel)`

I get a "view not found" error.

How to call that *view* from other controller

c#   asp.net-mvc

edited Dec 20 '16 at 17:25

🔲 Gone Coding
85.5k ● 23 ● 164 ● 181

asked Jan 13 '14 at 11:16

⬛ Vinit Patel
2,131 ● 3 ● 21 ● 45

3 Answers

Active Oldest Votes

▲

151

▼

✔

↺

Normally the views belong with a specific *matching controller* that supports its data requirements, or the view belongs in the `Views/Shared` folder if shared between controllers (hence the name).

"Answer" (but not recommended - see below):

You *can* refer to views/partial views from another controller, by specifying the full path (including extension) like:

```
return PartialView("~/views/ABC/XXX.cshtml", zyxmodel);
```

or a relative path (no extension), based on the answer by `@Max Toro`

```
return PartialView("../ABC/XXX", zyxmodel);
```

BUT THIS IS NOT A GOOD IDEA ANYWAY

*Note: These are *the only two syntax that work*. not `ABC\\XXX` or `ABC/XXX` or any other variation as those are all relative paths and do not find a match.

Better Alternatives:

You can use `Html.Renderpartial` in your view instead, but it requires the extension as well:

```
Html.RenderPartial("~/Views/ControllerName/ViewName.cshtml", modeldata);
```

Use `@Html.Partial` for inline Razor syntax:

```
@Html.Partial("~/Views/ControllerName/ViewName.cshtml", modeldata)
```

You can use the `../controller/view` syntax with no extension (again credit to @Max Toro):

```
@Html.Partial("../ControllerName/ViewName", modeldata)
```

Note: Apparently `RenderPartial` is slightly faster than Partial, but that is not important.

If you want to actually call the other controller, use:

```
@Html.Action("action", "controller", parameters)
```

Recommended solution: @Html.Action

My personal preference is to use `@Html.Action` as it allows each controller to manage its own views, rather than cross-referencing views from other controllers (which leads to a large spaghetti-like mess).

You would normally pass just the required key values (like any other view) e.g. for your example:

```
@Html.Action("XXX", "ABC", new {id = model.xyzId })
```

This will execute the `ABC.XXX` action and render the result in-place. This allows the views and controllers to remain separately self-contained (i.e. reusable).

Update Sep 2014:

I have just hit a situation where I could not use @Html.Action, but needed to create a view path based on a `action` and `controller` names. To that end I added this simple `View` extension method to `UrlHelper` so you can say return `PartialView(Url.View("actionName", "controllerName"), modelData)` :

```csharp
public static class UrlHelperExtension
{
    /// <summary>
    /// Return a view path based on an action name and controller name
    /// </summary>
    /// <param name="url">Context for extension method</param>
    /// <param name="action">Action name</param>
    /// <param name="controller">Controller name</param>
    /// <returns>A string in the form "~/views/{controller}/{action}.cshtml</returns>
    public static string View(this UrlHelper url, string action, string controller)
    {
        return string.Format("~/Views/{1}/{0}.cshtml", action, controller);
    }
}
```

edited Dec 5 '16 at 10:03   answered Jan 13 '14 at 14:21

GONE
AWAY   **Gone Coding**
        85.5k ● 23 ● 164 ● 181

---

3   @Html.Action is the way to go – Arman Oct 17 '16 at 6:45

Your recommended approach would make me call services from view to retrieve model my partial requires. Which I wan't to avoid. – Raimonds Apr 29 '19 at 11:37

The control searches for a view in the following order:

11

- First in shared folder
- Then in the folder matching the current controller (in your case it's Views/DEF)

As you do not have `xxx.cshtml` in those locations, it returns a "view not found" error.

Solution: You can use the complete path of your view:

Like

```
PartialView("~/views/ABC/XXX.cshtml", zyxmodel);
```

edited Jun 13 '14 at 13:36   answered Jan 13 '14 at 11:24

Gone Coding   **Arun Chandran C**
85.5k ● 23 ● 164 ● 181   **3,266** ● 6 ● 27 ● 43

@TrueBlueAussie thanks for correction. You explained well. Its not about earning `"best answer tick' and up-votes` just sharing .... Any way thanks for your point. :) – Arun Chandran C Jun 13 '14 at 13:31

Indeed. That's why I have editing your entry as well to improve it. – Gone Coding Jun 13 '14 at 13:36

6

Simply you could use:

```
PartialView("../ABC/XXX")
```

edited Mar 31 '15 at 18:28   answered Mar 31 '15 at 16:33

Pankaj Parkar   **Max Toro**
123k ● 18 ● 198 ● 260   **26.4k** ● 10 ● 68 ● 106

I was initially sceptical of this syntax, but it works just fine, including from default controller actions. Do you have a reference to go with it? – Gone Coding Jul 20 '15 at 18:18