

Returning Multiple partial views from single Controller action?

Asked 8 years, 10 months ago Active 2 years, 10 months ago Viewed 72k times



33



17



I need to update Multiple from an Ajax call , I am confused as in how to return these Multiple views from the Controller Action method.

asp.net-mvc-3

asked Aug 4 '11 at 6:38



Nikshep

2,019 ● 4 ● 19 ● 29

What are you trying to achieve? – [Bertrand Marron](#) Aug 4 '11 at 6:42

I have a page with two partial views one which show a chart and the other a list of item which is represented in a chart they are made in two different partial views. Now for a different grouping both these values would change ,an obvious work around might be to make them into a single partial view which I want to avoid. As independently they can be used more often in my app. – [Nikshep](#) Aug 4 '11 at 6:45

Why don't you simply make two ajax calls? – [frennky](#) Aug 4 '11 at 10:59

1 I have to implement that solution now but I would have preferred single call to the server ,but I am not able to find a simple way to do this. Thanks – [Nikshep](#) Aug 4 '11 at 11:10 ✎

3 Answers

Active

Oldest

Votes



53



You can only return one value from a function so you can't return multiple partials from one action method.

If you are trying to return two models to one view, create a view model that contains both of the models that you want to send, and make your view's model the new ViewModel. E.g.

Your view model would look like:



```
public class ChartAndListViewModel
{
    public List<ChartItem> ChartItems {get; set;};
}
```



```
public List<ListItem> ListItems {get; set;};  
}
```

Then your controller action would be:

```
public ActionResult ChartList()  
{  
    var model = new ChartAndListViewModel();  
    model.ChartItems = _db.getChartItems();  
    model.ListItems = _db.getListItems();  
  
    return View(model);  
}
```

And finally your view would be:

```
@model Application.ViewModels.ChartAndListViewModel  
  
<h2>Blah</h2>  
  
@Html.RenderPartial("ChartPartialName", model.ChartItems);  
@Html.RenderPartial("ListPartialName", model.ListItems);
```

edited Aug 4 '11 at 7:10

answered Aug 4 '11 at 6:41



dnatoli

6,592 ● 7 ● 49 ● 92



There is a very good example here....

<http://rhamesconsulting.com/2014/10/27/mvc-updating-multiple-partial-views-from-a-single-ajax-action/>

24

Create a helper method to package up the partial view...



```
public static string RenderRazorViewToString(ControllerContext controllerContext,  
    string viewName, object model)  
{  
    controllerContext.Controller.ViewData.Model = model;  
  
    using (var stringWriter = new StringWriter())  
    {
```

```

        var viewResult = ViewEngines.Engines.FindPartialView(controllerContext,
viewName);
        var viewContext = new ViewContext(controllerContext, viewResult.View,
controllerContext.Controller.ViewData, controllerContext.Controller.TempData,
stringWriter);
        viewResult.View.Render(viewContext, stringWriter);
        viewResult.ViewEngine.ReleaseView(controllerContext, viewResult.View);
        return stringWriter.GetStringBuilder().ToString();
    }
}

```

Create a controller action to bundle the multiple partial views....

```

[HttpPost]
public JsonResult GetResults(int someExampleInput)
{
    MyResultsModel model = CalculateOutputData(someExampleInput);

    var totalValuesPartialView = RenderRazorViewToString(this.ControllerContext,
"_TotalValues", model.TotalValuesModel);
    var summaryValuesPartialView = RenderRazorViewToString(this.ControllerContext,
"_SummaryValues", model.SummaryValuesModel);

    return Json(new { totalValuesPartialView, summaryValuesPartialView });
}

```

Each partial view can use its own model if required or can be bundled into the same model as in this example.

Then use an AJAX call to update all the sections in one go:

```

$('#getResults').on('click', function () {

    $.ajax({
        type: 'POST',
        url: "/MyController/GetResults",
        dataType: 'json',
        data: {
            someExampleInput: 10
        },
        success: function (result) {
            if (result != null) {
                $("#totalValuesPartialView").html(result.totalValuesPartialView);
                $("#summaryValuesPartialView").html(result.summaryValuesPartialView);
            } else {
                alert('Error getting data.');
```

```
    }  
    },  
    error: function () {  
        alert('Error getting data.');    }  
    });  
});
```

If you want to use this method for a GET request, you need to remove the `[HttpPost]` decorator and add `JsonRequestBehavior.AllowGet` to the returned `JsonResult` :

```
return Json(new { totalValuesPartialView, summaryValuesPartialView },  
    JsonRequestBehavior.AllowGet);
```

edited Aug 11 '17 at 7:16

answered Jan 23 '16 at 20:49



SharpC

4,358 ● 2 ● 33 ● 31

2 I like this solution because it allows you to return multiple partial views that don't exist within the same location on the page. – [kerl](#) Jan 19 '18 at 2:56

Maybe this solution can help you:

5

<http://www.codeproject.com/Tips/712187/Returning-More-Views-in-an-ASP-NET-MVC-Action>

answered Feb 17 '14 at 10:06



[user3318614](#)

51 ● 1 ● 1

Nice one, but only works for asp.net framework. Doesn't apply for asp.net core. ./ – [Antoine Meltzheim](#) Aug 6 '19 at 20:28
