



ASP.NET MVC 5 - Multiple View Models In Single View

This article is about implementing multiple view models in a single view via ASP.NET MVC5 platform.



Asma Khalid

Apr 12 2017

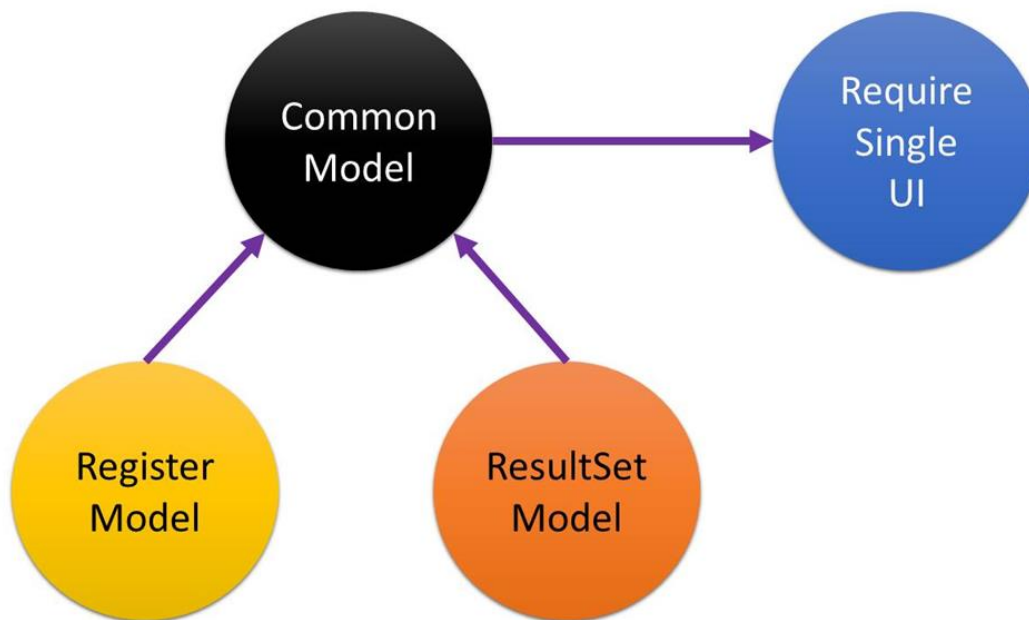
7

12

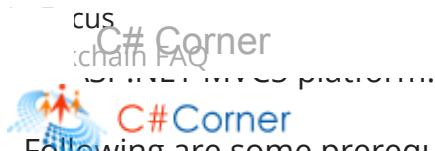
69.8k

[MultiModelSingleView.zip](#)

[Download Free .NET & JAVA Files API](#)



There is a common scenario, which is being asked quite a lot in ASP.NET MVC platforms. Thus, I have decided to demonstrate my take on this particular scenario. The scenario is as follows; i.e., create a single page, which will display the data from two different view models. Now, the question that arises here is if the ASP.NET MVC platform only attaches a single model to a single view, how can this be achieved? Let me demonstrate how this is done.

[ASK A QUESTION](#)[CONTRIBUTE](#)

Following are some prerequisites before you proceed further in this tutorial:

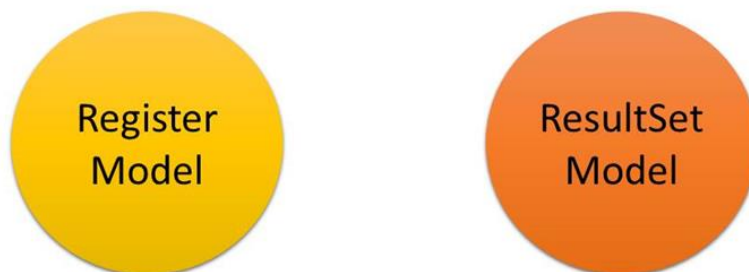
2. Knowledge of [HTML](#).
3. Knowledge of [JavaScript](#).
4. Knowledge of [Bootstrap](#).
5. Knowledge of [jQuery](#).
6. Knowledge of [C# programming](#).

You can download the complete source code for this tutorial or you can follow the step by step discussion below. The sample code is being developed in Microsoft Visual Studio 2015 Enterprise.

Before we jump into the technical working of this scenario, let's build a little conceptual understanding of how will we achieve this scenario in an ASP.NET MVC5 platform. To do achieve this, we will proceed in our code, as shown below i.e.

Step 1

We have two independent models Register Model & ResultSet Model



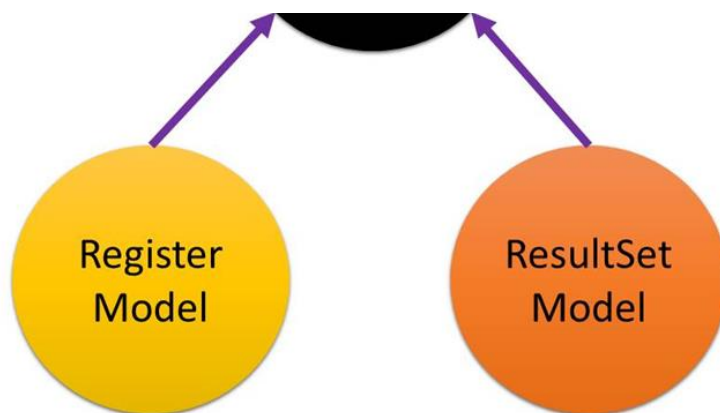
Step 2

Since we know the constraint on ASP.NET MVC platform i.e. we can only attach a single model to a single view. So, here we will create a parent place holder kind of model; i.e., Common Model, and make our actual view models the child to this model, as shown below

Common

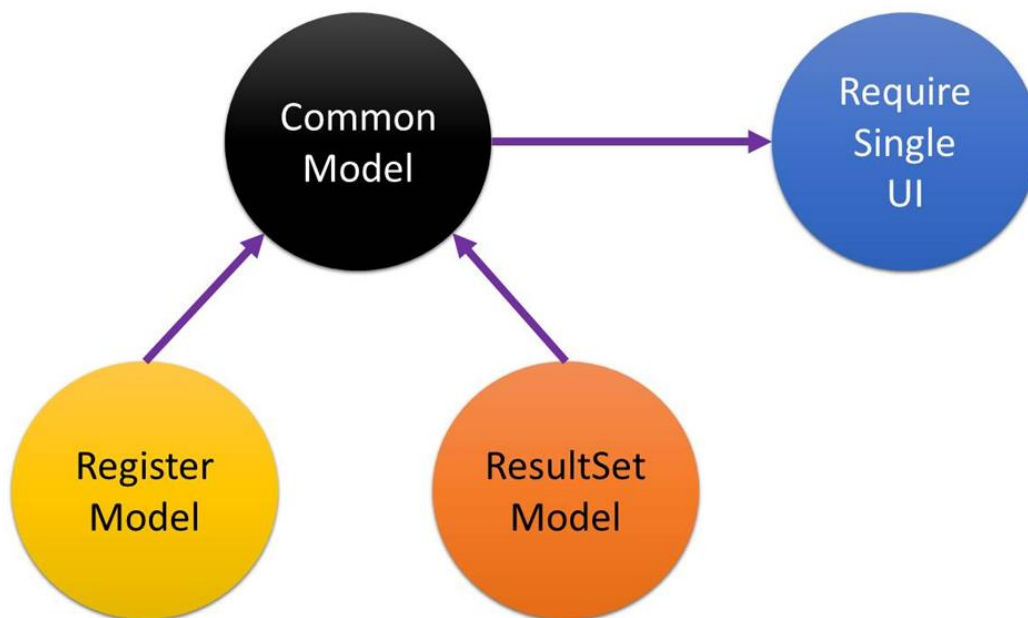
ASK A QUESTION

CONTRIBUTE



Step 3

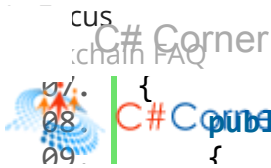
After combining our two separate models into a single model, we then can easily attach our common place holder model to our single view, as shown below.



Now, let's see, how can we code this.

1. Create a new MVC Web project and name it MultiModelSingleView.
2. Create a file LoginObj.cs under Models folder and replace the code given below in it

```
01. using System;  
02. using System.Collections.Generic;  
03. using System.Linq;  
04. using System.Web;
```



```
{  
    public class LoginObj  
    {
```

[ASK A QUESTION](#)[CONTRIBUTE](#)

```
12.         public string Username { get; set; }  
13.  
14.         public string Password { get; set; }  
15.     }  
16. }
```

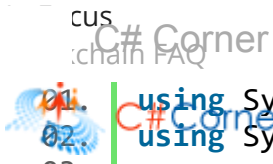
Now, create our view models files -- AccountViewModel.cs, ResultSetViewModel.cs & CommonViewModel.cs -- under Models folder and replace the code given below in each file accordingly, as shown below

Replace the code given below in AccountViewModel.cs file

```
01. using System.Collections.Generic;  
02. using System.ComponentModel.DataAnnotations;  
03.  
04. namespace MultiModelSingleView.Models  
05. {  
06.     public class AccountViewModel  
07.     {  
08.         [Display(Name = "Id")]  
09.         public int Id { get; set; }  
10.  
11.         [Display(Name = "Enter Username")]  
12.         public string Username { get; set; }  
13.  
14.         [Display(Name = "Enter Password")]  
15.         public string Password { get; set; }  
16.     }  
17. }
```

Replace the code given below in ResultSetViewModel.cs file i.e.

```
01. using System.Collections.Generic;  
02. using System.ComponentModel.DataAnnotations;  
03.  
04. namespace MultiModelSingleView.Models  
05. {  
06.     public class ResultSetViewModel  
07.     {  
08.         [Display(Name = "Result")]  
09.         public List<LoginObj> ResultSet { get; set; }  
10.     }  
11. }
```



```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
```

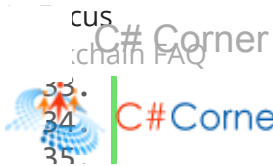
[ASK A QUESTION](#)
[CONTRIBUTE](#)

```
05.
06.     public class CommonViewModel
07.     {
08.         public AccountViewModel AccountVM { get; set; }
09.         public ResultSetViewModel ResultSetVM { get; set; }
10.     }
11. }
```

In all the snippets given above, we have created our view models according to the conceptual understanding, which we have developed.

Create a new controller file AccountController.cs under Controllers folder and replace the code given below in it

```
01.  //-----
02.  // <copyright file="HomeController.cs" company="None">
03.  //      Copyright (c) Allow to distribute this code.
04.  // </copyright>
05.  // <author>Asma Khalid</author>
06.  //-----
07.  -
08.  namespace MultiModelSingleView.Controllers
09.  {
10.      using System;
11.      using System.Globalization;
12.      using System.Linq;
13.      using System.Security.Claims;
14.      using System.Threading.Tasks;
15.      using System.Web;
16.      using System.Web.Mvc;
17.      using Microsoft.AspNet.Identity;
18.      using Microsoft.AspNet.Identity.Owin;
19.      using Microsoft.Owin.Security;
20.      using MultiModelSingleView.Models;
21.      using System.Collections.Generic;
22.      using System.IO;
23.      using System.Reflection;
24.
25.      public class AccountController : Controller
26.      {
27.
28.          #region Register method
29.
30.          #region GET: /Account/Register
```



```
// GET: /Account/Register
[AllowAnonymous]
public ActionResult Register()
```

[ASK A QUESTION](#)
[CONTRIBUTE](#)

```
38.         CommonViewModel model = new CommonViewModel();
39.         model.AccountVM = new AccountViewModel();
40.         model.ResultSetVM = new ResultSetViewModel();
41.
42.         // Get Result
43.         model.ResultSetVM.ResultSet = this.LoadData();
44.
45.         return View(model);
46.     }
47.
48. #endregion
49.
50. #region POST: /Account/Register
51.
52. //
53. // POST: /Account/Register
54. [HttpPost]
55. [AllowAnonymous]
56. [ValidateAntiForgeryToken]
57. public ActionResult Register(CommonViewModel model)
58. {
59.     if (ModelState.IsValid)
60.     {
61.         // Inserting.
62.         this.StoreData(model.AccountVM.Username, model.AccountVM.F
63.
64.         // Get Result
65.         model.ResultSetVM = new ResultSetViewModel();
66.         model.ResultSetVM.ResultSet = this.LoadData();
67.     }
68.
69.     // If we got this far, something failed, redisplay form
70.     return View(model);
71. }
72.
73. #endregion
74.
75. #endregion
76.
77. #region Helpers
78.
79. #region Load Data
80.
81. /// <summary>
82. /// Load data method.
83. /// </summary>
84. /// <returns>Returns - Data</returns>
```



// Initialization.

List<LoginObj> lst = new

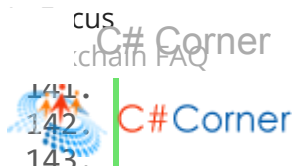
ASK A QUESTION

CONTRIBUTE

```

92.         // Initialization.
93.         string line = string.Empty;
94.         string srcFilePath = "content/files/login_list.txt";
95.         var rootPath = Path.GetDirectoryName(Assembly.GetExecutingAssembly().ManifestResourceStream(srcFilePath).Path);
96.         var fullPath = Path.Combine(rootPath, srcFilePath);
97.         string filePath = new Uri(fullPath).LocalPath;
98.         StreamReader sr = new StreamReader(new FileStream(filePath, FileMode.Open, FileAccess.Read));
99.
100.        // Read file.
101.        while ((line = sr.ReadLine()) != null)
102.        {
103.            // Initialization.
104.            LoginObj infoObj = new LoginObj();
105.            string[] info = line.Split(',');
106.
107.            // Setting.
108.            infoObj.Id = Convert.ToInt32(info[0].ToString());
109.            infoObj.Username = info[1].ToString();
110.            infoObj.Password = info[2].ToString();
111.
112.            // Adding.
113.            lst.Add(infoObj);
114.        }
115.
116.        // Closing.
117.        sr.Dispose();
118.        sr.Close();
119.    }
120.    catch (Exception ex)
121.    {
122.        // info.
123.        Console.WriteLine(ex);
124.    }
125.
126.    // info.
127.    return lst;
128.}
129.
130.#endregion
131.
132.#region Store Data
133.
134./// <summary>
135./// Store data method.
136./// </summary>
137.private void StoreData(string username, string password)
138.{

```



ASK A QUESTION

CONTRIBUTE

```

146.         ? (this.LoadData().OrderByDescending(p => p.Id
147.             : 1;
148.
149.         // Initialization.
150.         string line = string.Empty;
151.         string srcFilePath = "content/files/login_list.txt";
152.         var rootPath = Path.GetDirectoryName(Assembly.GetExecuting
153.         var fullPath = Path.Combine(rootPath, srcFilePath);
154.         string filePath = new Uri(fullPath).LocalPath;
155.         StreamWriter sw = new StreamWriter(new FileStream(filePath
156.
157.         // Write file.
158.         string content = idVal.ToString() + "," + username + "," -
159.         sw.WriteLine(content);
160.
161.         // Closing.
162.         sw.Dispose();
163.         sw.Close();
164.     }
165.     catch (Exception ex)
166.     {
167.         // info.
168.         Console.Write(ex);
169.     }
170. }
171.
172. #endregion
173.
174. #endregion
175. }
176. }

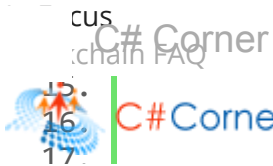
```

Let's break down the code given above, method by method

```

01. #region Load Data
02.
03. /// <summary>
04. /// Load data method.
05. /// </summary>
06. /// <returns>Returns - Data</returns>
07. private List<LoginObj> LoadData()
08. {
09.     // Initialization.
10.     List<LoginObj> lst = new List<LoginObj>();
11.
12.     try

```

[ASK A QUESTION](#)
[CONTRIBUTE](#)

```

15. string line = string.Empty;
16. string srcFilePath = "conte
17. var rootPath = Path.GetDirectoryName(Assembly.GetExecutingAssembly

20. StreamReader sr = new StreamReader(new FileStream(filePath, FileMode
21.
22. // Read file.
23. while ((line = sr.ReadLine()) != null)
24. {
25.     // Initialization.
26.     LoginObj infoObj = new LoginObj();
27.     string[] info = line.Split(',');
28.
29.     // Setting.
30.     infoObj.Id = Convert.ToInt32(info[0].ToString());
31.     infoObj.Username = info[1].ToString();
32.     infoObj.Password = info[2].ToString();
33.
34.     // Adding.
35.     lst.Add(infoObj);
36. }
37.
38. // Closing.
39. sr.Dispose();
40. sr.Close();
41. }
42. catch (Exception ex)
43. {
44.     // info.
45.     Console.Write(ex);
46. }
47.
48. // info.
49. return lst;
50. }
51.
52. #endregion

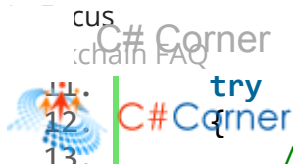
```

The code given above creates a LoadData() method, which will load the data from a file, if the file contains any data, initially, the file is empty, since we have not registered any account.

```

01. #region Store Data
02.
03. /// <summary>
04. /// Store data method.
05. /// </summary>
06. private void StoreData(string username, string password)
07. {
08.     // Initialization.

```



ASK A QUESTION

CONTRIBUTE

```

11. // Setting
12.
13.
14.
15.
16.         : 1;
17.
18.         // Initialization.
19.         string line = string.Empty;
20.         string srcFilePath = "content/files/login_list.txt";
21.         var rootPath = Path.GetDirectoryName(Assembly.GetExecutingAssembly);
22.         var fullPath = Path.Combine(rootPath, srcFilePath);
23.         string filePath = new Uri(fullPath).LocalPath;
24.         StreamWriter sw = new StreamWriter(new FileStream(filePath, FileMode
25.
26.         // Write file.
27.         string content = idVal.ToString() + "," + username + "," + password;
28.         sw.WriteLine(content);
29.
30.         // Closing.
31.         sw.Dispose();
32.         sw.Close();
33.     }
34.     catch (Exception ex)
35.     {
36.         // info.
37.         Console.Write(ex);
38.     }
39. }
40.
41. #endregion

```

Now, the code given above creates a StoreData(...) method, which will store the data into the file, as we register any new account.

```

01. #region GET: /Account/Register
02.
03.     //
04.     // GET: /Account/Register
05.
06.     [AllowAnonymous]
07.     public ActionResult Register()
08.     {
09.         // Initialization.
10.         CommonViewModel model = new CommonViewModel();
11.         model.AccountVM = new AccountViewModel();
12.         model.ResultSetVM = new ResultSetViewModel();
13.
14.         // Get Result
15.         model.ResultSetVM.ResultSet = this.LoadData();

```



#endregion

ASK A QUESTION

CONTRIBUTE

data to ResultSetViewModel. Notice here that now in our view, instead of returning an individual model to the view, we are returning our common view model, since we have attached that with our View.

```

01. #region POST: /Account/Register
02.
03.     //
04.     // POST: /Account/Register
05.     [HttpPost]
06.     [AllowAnonymous]
07.     [ValidateAntiForgeryToken]
08.     public ActionResult Register(CommonViewModel model)
09.     {
10.         if (ModelState.IsValid)
11.         {
12.             // Inserting.
13.             this.StoreData(model.AccountVM.Username, model.AccountVM.P
14.
15.             // Get Result
16.             model.ResultSetVM = new ResultSetViewModel();
17.             model.ResultSetVM.ResultSet = this.LoadData();
18.         }
19.
20.         // If we got this far, something failed, redisplay form
21.         return View(model);
22.     }
23.
24. #endregion

```

The code given above creates our view method Register() HTTP POST, which will simply load our data to ResultSetViewModel after storing the register account details received via account view model. Notice here again that now in our view instead of returning an individual model to the view, we are returning our common view model, since we have attached that with our view.

Now, create a new view file Register.cshtml under Views\Account folder and replace the code given below in it

```

01. @model MultiModelSingleView.Models.CommonViewModel
02. @{
03.     ViewBag.Title = "Register";
04. }
05.
06. <h2>@ViewBag.Title.</h2>
07.

```


[ASK A QUESTION](#)
[CONTRIBUTE](#)

```

11.     @Html.AntiForgeryToken()
12.     <h4>Create a new account </h4>
13.
14.     <div class="form-group">
15.         @Html.LabelFor(m => m.AccountVM.Username, new { @class = "col-
16. md-2 control-label" })
17.         <div class="col-md-10">
18.             @Html.TextBoxFor(m => m.AccountVM.Username, new { @class = "fo
19. control" })
20.         </div>
21.     </div>
22.     <div class="form-group">
23.         @Html.LabelFor(m => m.AccountVM.Password, new { @class = "col-
24. md-2 control-label" })
25.         <div class="col-md-10">
26.             @Html.PasswordFor(m => m.AccountVM.Password, new { @class = "fo
27. control" })
28.         </div>
29.     </div>
30.     <div class="form-group">
31.         <div class="col-md-offset-2 col-md-10">
32.             <input type="submit" class="btn btn-
33. default" value="Register" />
34.         </div>
35.     </div>
36. }
37.
38. <h2>Result List</h2>
39.
40. @if (Model.ResultSetVM.ResultSet != null)
41. {
42.     for (int i = 0; i < Model.ResultSetVM.ResultSet.Count; i++)
43.     {
44.         <div class="row">
45.             <div class="col-md-2">
46.                 <p>@Model.ResultSetVM.ResultSet[i].Id</p>
47.             </div>
48.             <div class="col-md-2">
49.                 <p>@Model.ResultSetVM.ResultSet[i].Username</p>
50.             </div>
51.             <div class="col-md-2">
52.                 <p>@Model.ResultSetVM.ResultSet[i].Password</p>
53.             </div>
54.         </div>
55.     }
56. }
57.
58. @section Scripts {
59.     @Scripts.Render("~/bundles/jqueryval")
60. }

```



ASK A QUESTION

CONTRIBUTE

In the code given above, we have created a single

Create a our standard view layout file `_Layout.cshtml` under `Views\Shared` folder. Replace the code given below in it i.e.

```

01. <!DOCTYPE html>
02. <html>
03. <head>
04.     <meta charset="utf-8" />
05.     <meta name="viewport" content="width=device-width, initial-
scale=1.0">
06.     <title>@ViewBag.Title</title>
07.     @Styles.Render("~/Content/css")
08.     @Scripts.Render("~/bundles/modernizr")
09.
10.     <!-- Font Awesome -->
11.     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.4.0/css/font-awesome.min.css" />
12. </head>
13. <body>
14.     <div class="navbar navbar-inverse navbar-fixed-top">
15.         <div class="container">
16.             <div class="navbar-header">
17.                 <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
18.                     <span class="icon-bar"></span>
19.                     <span class="icon-bar"></span>
20.                     <span class="icon-bar"></span>
21.                 </button>
22.             </div>
23.         </div>
24.     </div>
25.     <div class="container body-content">
26.         @RenderBody()
27.         <hr />
28.         <footer>
29.             <center>
30.                 <p>
<strong>Copyright © @DateTime.Now.Year - <a href="http://www.asmak9.com/">
</strong> All rights reserved.</p>
31.             </center>
32.         </footer>
33.     </div>
34.
35.     @Scripts.Render("~/bundles/jquery")
36.     @Scripts.Render("~/bundles/bootstrap")
37.

```

[ASK A QUESTION](#)[CONTRIBUTE](#)

shown below.

A screenshot of a web browser window displaying a registration form. The browser's address bar shows 'localhost:23697'. The page has a dark blue header with the title 'Register.' and the subtitle 'Create a new account.' Below the header, there are two input fields: 'Enter Username' and 'Enter Password'. A 'Register' button is positioned below the password field. Underneath the registration form, there is a section titled 'Result List'. At the bottom of the page, a copyright notice reads 'Copyright © 2017 - Asma's Blog. All rights reserved.' The browser's taskbar at the bottom shows several open tabs, including 'Most Visited', 'Getting Started', 'WP Engine Status', 'Trouble editing home ...', 'YouTube to MP3 Con...', 'WatchFreeMovies | Le...', '20 Psycho-Pass Quote...', 'Download MediaTek ...', and 'Assassination Classro...'. The system clock in the bottom right corner shows '3:15 PM'.

CUS
C# Corner
chat FAQ

ASK A QUESTION

CONTRIBUTE

Most Visited Getting Started WP Engine Status Trouble editing home ... YouTube to MP3 Con... WatchFreeMovies | Le... 20 Psycho-Pass Quote... Download MediaTek ... Assassination Classro...

Register.

Create a new account.

Enter Username

Enter Password

Result List

1	Test-1	123456
2	Test-2	gj24525
3	Test-4	grgk59045

Copyright © 2017 - Asma's Blog. All rights reserved.

Conclusion

In this article, you learned about mapping multiple view models to a single UI view in an ASP.NET MVC5 platform. You also learned about conceptual understanding behind this scenario along with how to achieve this scenario from the coding perspective.

ASP.NET

MVC



Asma Khalid *TOP 500*

Computer Programmer by Profession, Computer Scientist by Heart, Fanatic Explorer, Technology Centric. I am a Versatile Computer Science Evangelist. Enjoy doodling with technology.

<http://www.asmak9.com/>

191

2.2m

3

2

7

12



Type your comment here and press Enter Key (Minimum 10 characters)

I am having errors in html page like (name 'model' etc doesn't exit in current context) what should i



ASK A QUESTION

CONTRIBUTE



Anmol Butt

1757 6 0

Dec 07, 2018

0



How to connect this with database.... please help me out in this also???

Anmol Butt

1757 6 0

Dec 07, 2018

0

Look into this <https://bit.ly/2Sx78U9>

Asma Khalid

191 9.6k 2.2m

Dec 07, 2018

0



What if i want to use paging in the index method how do i do that

yusuf ogundiji

1758 5 0

Jul 04, 2018

0

0 Reply



Nice article Asma. Alternate method is also used through Expando object.

Dilshad Ali

1216 559 10.8k

Apr 13, 2017

0

0 Reply



Thank you all for your support.

Asma Khalid

191 9.6k 2.2m

Apr 13, 2017

1

0 Reply



I have vs 2013 installed how can i convert or run in vs 2013 ??

Manav Pandya

85 19.7k 604.4k

Apr 13, 2017

0

1 Reply



[Manav Pandya](#) you can import .csproj file into VS 2013 instead of directly opening the .snl file. Hope this helps

Asma Khalid

191 9.6k 2.2m

Apr 13, 2017

0



Thanks for sharing ma'm

Manav Pandya

85 19.7k 604.4k

Apr 13, 2017

0

0 Reply



Nice article For start-up engineers

Hamid Khan

586 2.4k 175.2k

Apr 12, 2017

0

0 Reply

OUR TRAINING

Mastering React and TypeScript

[ASK A QUESTION](#)[CONTRIBUTE](#)

TRENDING

- 01 GraphQL In .NET Core Web API With Entity Framework Core - Part Three
- 02 C# Coding Standards 🧐
- 03 Why Agile Is Important In Software Development
- 04 ASP.NET MVC Request Life Cycle
- 05 CRUD Operations On CosmosDB Mongo API Using ASP.NET Core
- 06 Create Registration And Login Page Using Angular 7 And Web API
- 07 GraphQL In .NET Core Web API With Entity Framework Core - Part One
- 08 What Does "var" Mean In C#?
- 09 Upload Files In Azure Blob Storage Using ASP.NET Core
- 10 Best Programming Language for Mobile App Development

[View All](#) ○

[About Us](#) [Contact Us](#) [Privacy Policy](#) [Terms](#) [Media Kit](#) [Sitemap](#) [Report a Bug](#) [FAQ](#) [Partners](#)
[C# Tutorials](#)

©2019 C# Corner. All contents are copyright of their authors.