



in MVC

ASK A QUESTION



Abhimanyu K Vatsa

Oct 19 2012

Article

20

45

573.1k

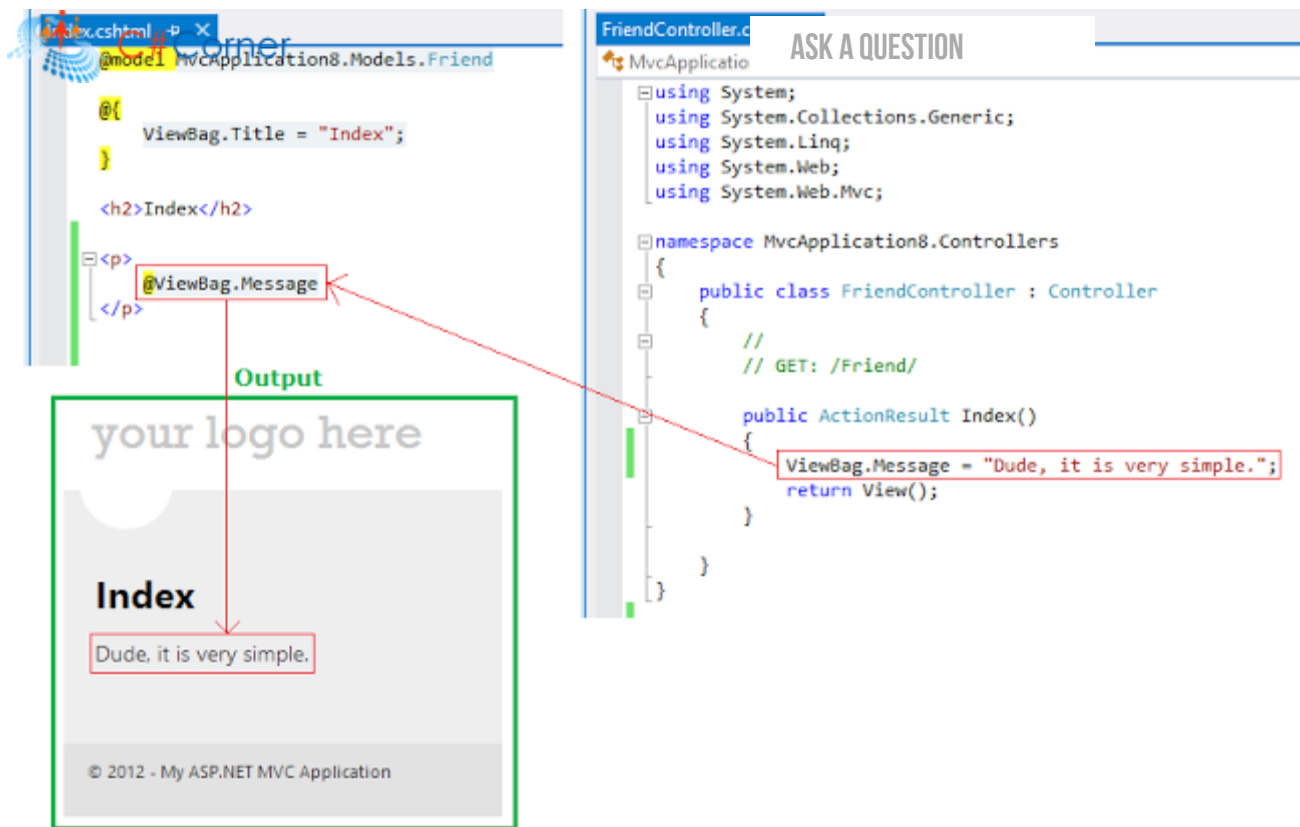
[Download 100% FREE Spire Office APIs](#)

There are various ways to pass data from a Controller to a View. I'm going to discuss how Controllers interact with Views and specifically cover ways you can pass data from a Controller to a View to render a response back to a client. So, let's get started.

ViewBag

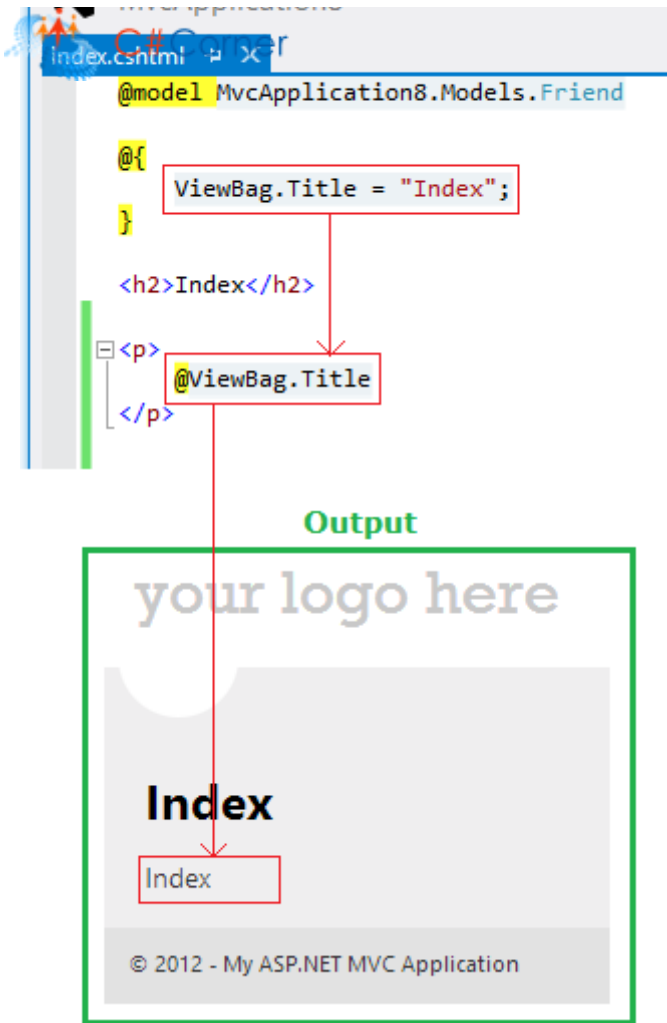
ViewBag is a very well known way to pass the data from Controller to View & even View to View. ViewBag uses the dynamic feature that was added in C# 4.0. We can say ViewBag=ViewData + Dynamic wrapper around the ViewData dictionary. Let's see how it is used.

Between Controller and View



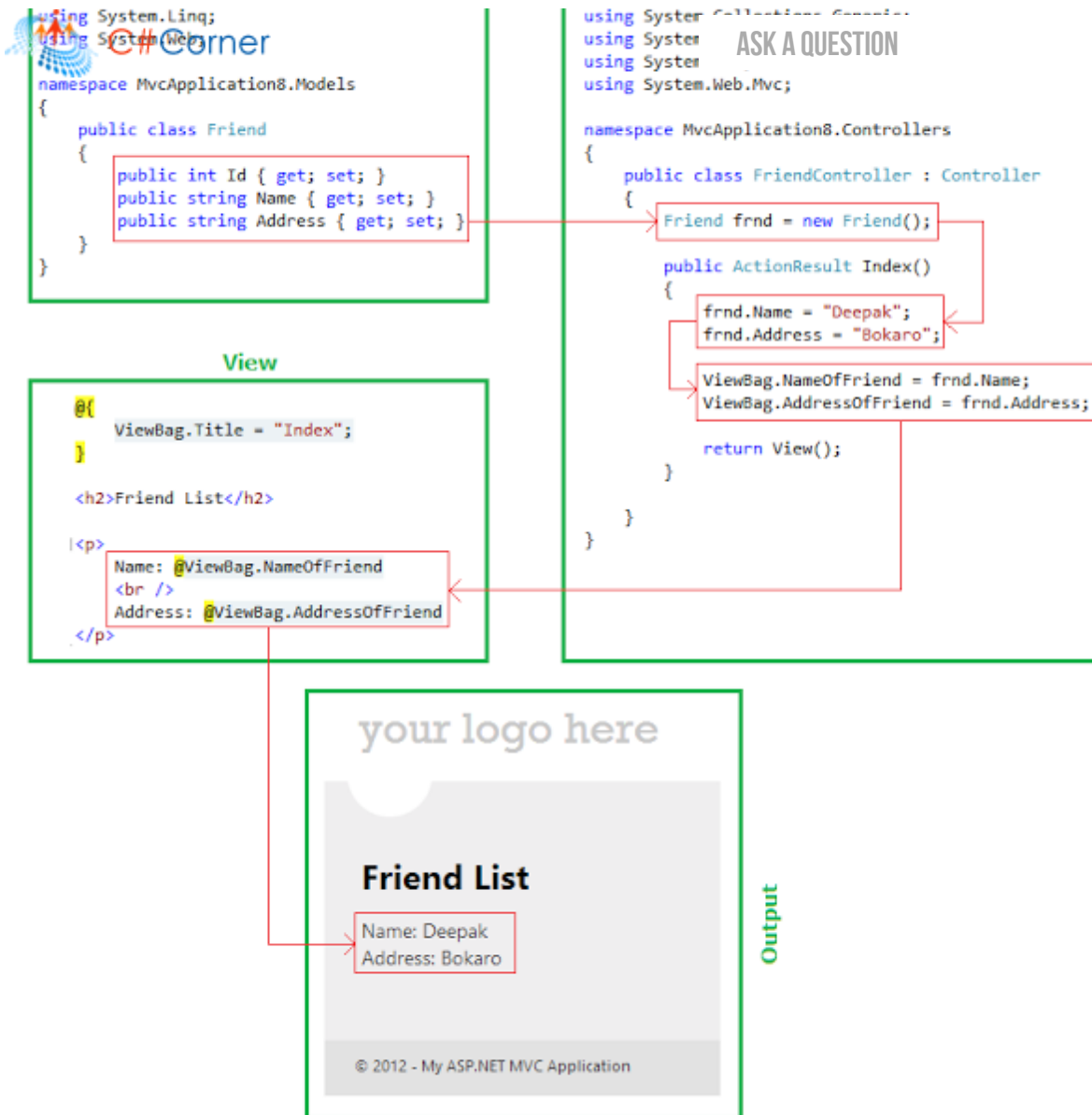
In the above image, you can see how data flows from the "Controller" to the "View", and how it looks in the browser.

Between View to View

[ASK A QUESTION](#)

In the above image, you see how data is initialized on the "View" page itself using `ViewBag.Title = "Index"` and then how it is getting rendered using `@ViewBag.Title`. What is "Title"? It is nothing more than a key, which has very limited availability and can be used on the same page only. So, the key naming is up to you, use any name which makes you happy.

Look at one more case, where I will take the advantage of "Model".



In the above case, we have a "Model" defined by name "Friend" that has three properties "Id", "Name" and "Address". On the "Controller", we have an object of the "Friend" class named "frnd" and then using the dot (.) operation properties are assigned then attached to these properties of the ViewBag.

Look at one more example, in which a list of students is passed using ViewBag.

The diagram illustrates the data flow in an MVC application. On the right, the **FriendController** (C#) is shown with the `Index()` action method. It creates a `List<string>` named `students` containing the names "Deepak Kumar", "Rohit Ranjan", and "Manish Kumar". It then assigns this list to `ViewBag.Students` and returns the view.

On the left, the **Index.cshtml** (View) is shown. It sets `ViewBag.Title = "Index";` and displays an

Friend List

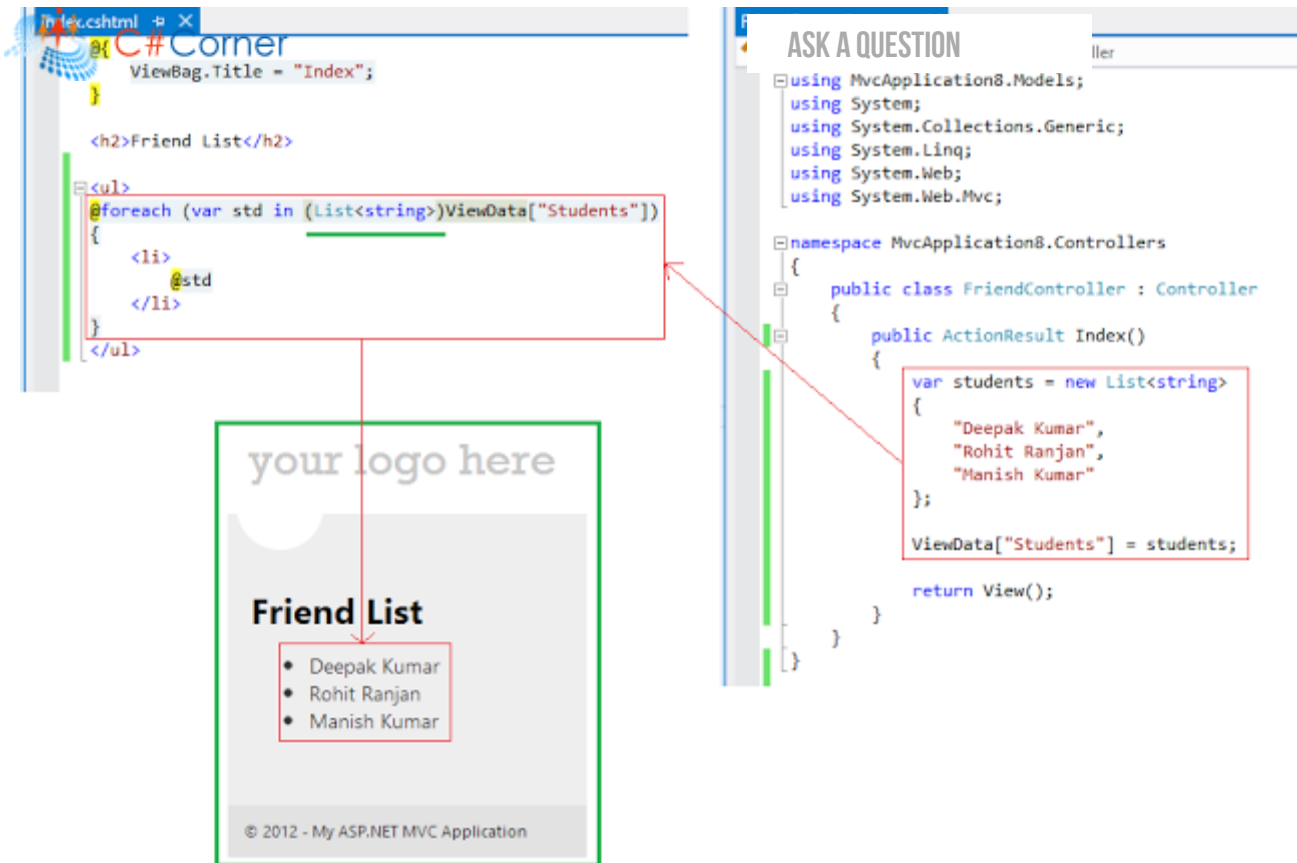
. It uses a `@foreach` loop to iterate over `ViewBag.Students`, displaying each name as a list item (`@std`).

Below the code, a screenshot of the rendered web page is shown. It features a placeholder for a logo, the title **Friend List**, and a bulleted list of the three names: Deepak Kumar, Rohit Ranjan, and Manish Kumar. A red arrow indicates the data flow from the controller's `students` list to the view's `@foreach` loop, and another red arrow points from the rendered list in the browser back to the view's loop.

So, in this way we can pass the list of students. I hope this is clear to you.

ViewData

ViewBag and ViewData serves the same purpose in allowing developers to pass data from controllers to views. When you put objects in either one, those objects become accessible in the view. Let's look at one example:

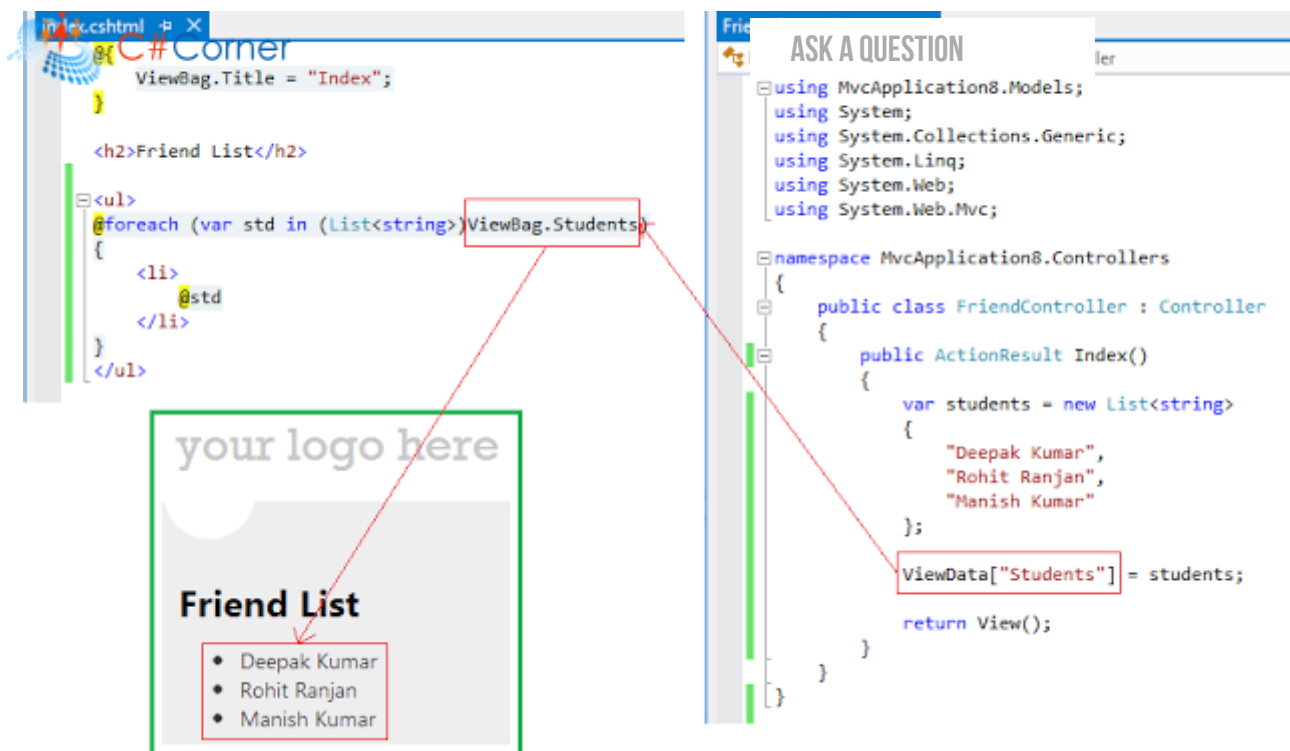


In the above image, everything is normal instead something in foreach loop. ViewData is typed as a dictionary containing "objects", we need to cast ViewData["Students"] to a List<string> or an IEnumerable<string> in order to use the foreach statement on it. Such as in:

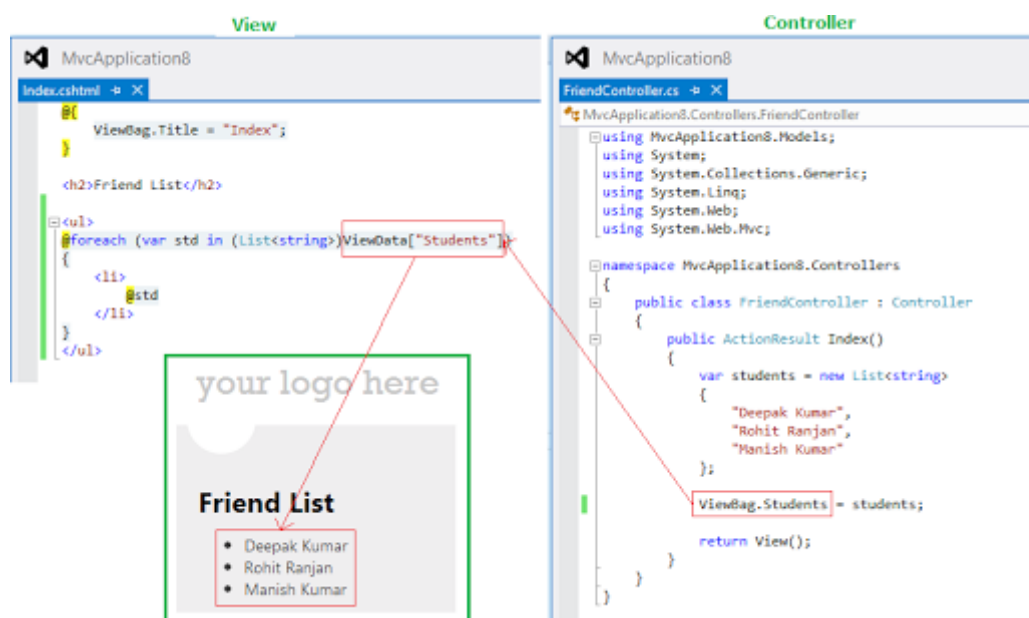
```
01. @foreach (var std in (List<string>)ViewData["Students"])
02. //OR
03. @foreach (var std in (IEnumerable<string>)ViewData["Students"])
```

Now look at one more beauty of MVC, you can put data into the ViewBag and access it from ViewData or put data in the ViewData and access it from the ViewBag, here you have all the freedom.

ViewData to ViewBag



ViewBag to ViewData



So these two (ViewBag and ViewData) things seem to work almost exactly the same. Then, what's the difference? The difference is only how you access the data. ViewBag is actually just a wrapper around the ViewData object, and its whole purpose is to let you use access the data dynamically instead of using magic <strings> conversion, you can realize it by the above examples. Some people prefer one style over the other. You can pick whichever makes you happy. In fact, because they're the same data just with two different ways of accessing it, you can use them interchangeably like ViewData to ViewBag or ViewBag to ViewData. It is not recommended, however, that you actually use them interchangeably, since it will will confuse others.

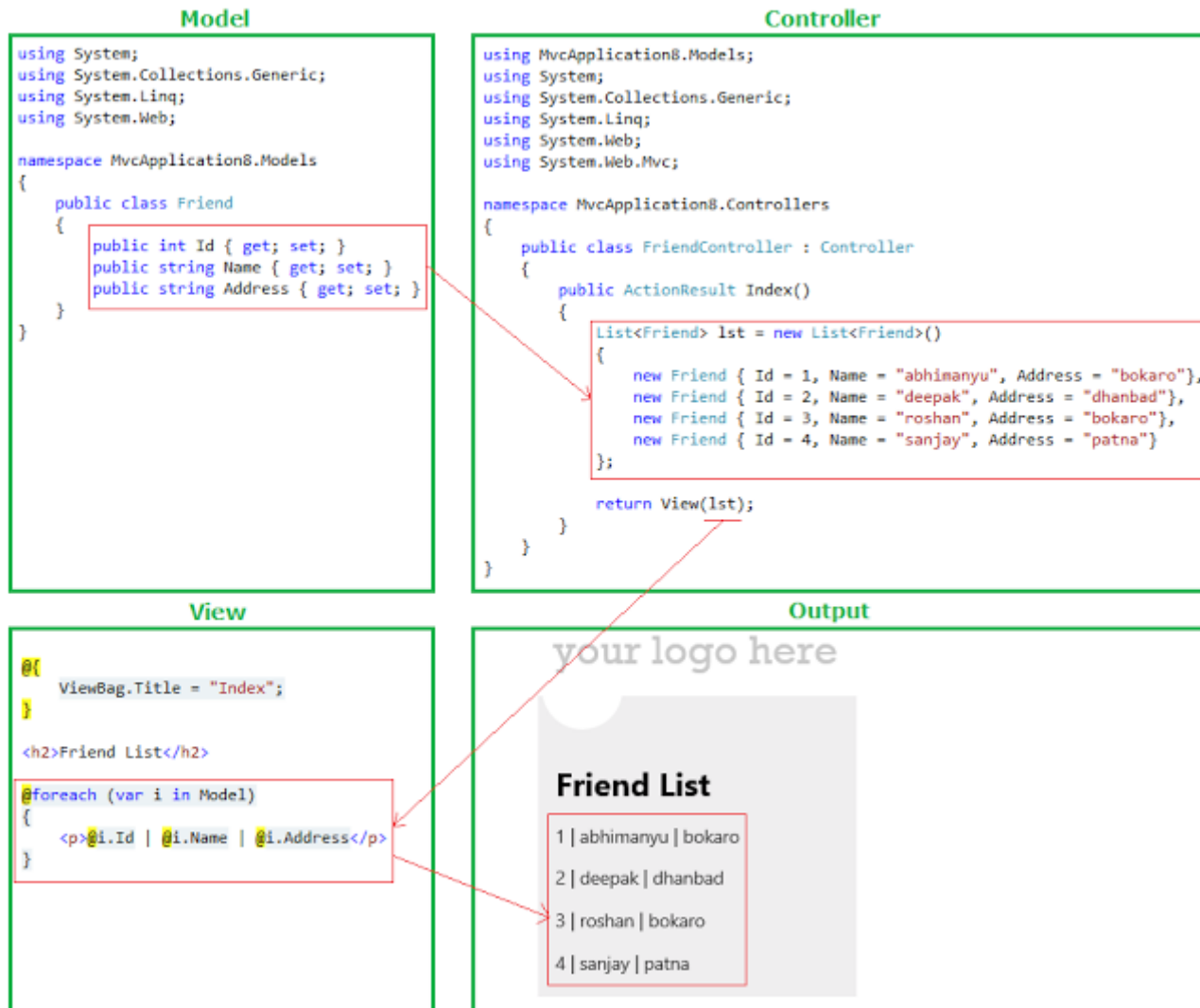
First Half of the Year MVPs Announced

data using a model also and this will provide you full int

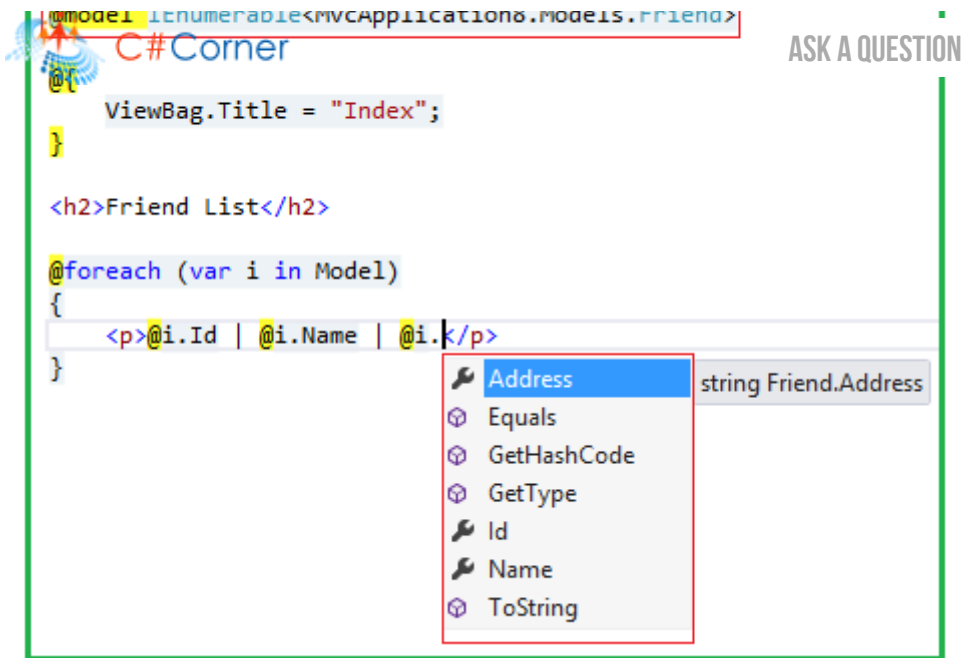
ASK A QUESTION

ViewModel

Using ViewModel we can also pass the data from the Controller to View; let's look at the image.



In the above image, we have multiple people in a list form being passed as a View, simple. I will add one more thing here, you are not going to get intellisense support or a strongly typed view page here, to get it do it this way. Just add a reference of the model by using the IEnumerable interface and you are done.



Please read this blog also:

TempData

TempData is meant to be a very short-lived instance, and you should only use it during the current and the subsequent requests only. Since TempData works this way, you need to know for sure what the next request will be, and redirecting to another view is the only time you can guarantee this. You can use TempData to pass error messages or something similar.

Example1: Using TempData like a ViewData and ViewBag.

```

01. public class FriendController : Controller
02. {
03.     //
04.     // GET: /Friend/
05.     public ActionResult Index()
06.     {
07.         ViewData["VDFriend"] = "Deepak K Gupta";
08.         ViewBag.VBFriend = "Deepak K Gupta";
09.         TempData["TDFriend"] = "Deepak K Gupta";
10.         return View();
11.     }
12. }

```

And on "View",

```

01. <p>Using ViewData: @ViewData["VDFriend"]</p>
02. <p>Using ViewBag: @ViewBag.VBFriend</p>
03. <p>Using TempData: @TempData["TDFriend"] </p>

```

This is a simple example, but we are not using the real advantage of TempData, let's look at one more example.



C# Corner

ASK A QUESTION

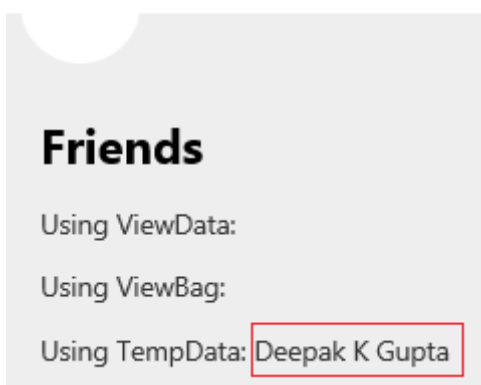
```
01. public class FriendController : Controller
02. {
03.     //
04.     // GET: /Friend/
05.
06.     public ActionResult Index()
07.     {
08.         ViewData["VDFriend"] = "Deepak K Gupta";
09.         ViewBag.VBFriend = "Deepak K Gupta";
10.         TempData["TDFriend"] = "Deepak K Gupta";
11.
12.         return new RedirectResult(@"~\Friend\AnotherPage\");
13.     }
14.
15.     public ActionResult AnotherPage()
16.     {
17.         return View();
18.     }
19.
20. }
```

And on "View":

```
01. <p>Using ViewData: @ViewData["VDFriend"]</p>
02. <p>Using ViewBag: @ViewBag.VBFriend</p>
03. <p>Using TempData: @TempData["TDFriend"]</p>
```

As in the above "FriendController", I'm redirecting the view, in other words the Index() view will be redirected to the AnotherPage() view instantly and now on another view after one redirect we won't be able to get data from ViewData or ViewBag but TempData will work here.

your logo here



Please read this blog also: <http://weblogs.asp.net/scottgu/archive/2007/12/06/asp-net-mvc-framework-part-3-passing-viewdata-from-controllers-to-views.aspx>.

Download 100% FREE Spire Office APIs

MVC

MVC Controller and View

pass data from Controller to View in MVC



Senior Software Developer | Microsoft MVP | Trainer | Speaker | Blogger at www.itorian.com

ASK A QUESTION

4 11m 2 5

[View Previous Comments](#)

20 45



Type your comment here and press Enter Key (Minimum 18 characters)



Very nicely composed and explained .. thanks

[Udit Sharma](#)

1400 45 213

Jun 15, 2017

0 0 Reply



Sorry to say that people do not expect a article on very basic and easy subject from you. write article on challenging area may be like elastic search, angular 2 etc

[tri_inn](#)

848 719 12.3k

May 26, 2017

0 1 Reply



Btw, this post was published on my blog in early 2011 and on c# corner in 2012. do you really see angular 2 a challenging area which i should write on?

[Abhimanyu K Vatsa](#)

4 62.6k 11m

May 26, 2017

1



Very very thanks. It can't be explained better than that

[Mansur Akbas](#)

1444 1 0

Apr 16, 2017

1 1 Reply



Great explanation sir !

[ar v](#)

1442 3 0

May 26, 2017

1



I must say this is one of the best article i have ever seen in dot net... sooooo specific and accurate.. and easy understandable for any beginner

[Bhavin Bhaskaran](#)

1444 1 0

Apr 10, 2017

0 0 Reply



Thank you. how to pass value div id view to controller in mvc5. please help me.

[veera muthu](#)

1408 37 0

Mar 27, 2017

0 0 Reply



Explained very nicely thank you.

[Saurabh Borode](#)

1444 1 0

Dec 15

0 0



can you please tell me how can i use the bool on in the function which is in the view.Thanks

Nadeem Siddiqui

1444 1 0

ASK A QUESTION

Dec 01, 2016

0 0 Reply



Nice...

kalu singh rao

217 6.1k 56.1k

Jul 11, 2016

0 0 Reply



Thanks, will be very useful for beginners.

Chocks

1440 5 1.5k

Mar 31, 2016

0 0 Reply



Nice One

Gowtham K

88 14.5k 1.1m

Dec 14, 2015

0 0 Reply

Comment Using



C#Corner

ASK A QUESTION

Add a comment...



Sachin Giri · Senior Software Engineer at Gateway Group of Companies
great post .. All the ways are performed with beauty .

Like · Reply · Sep 22, 2015 1:54am



Trong Tran · Works at AsNet Co., Ltd.
It look easy for newbie MVC. Thanks a lot.

Like · Reply · 1 · Dec 13, 2015 11:10pm



HD Bilişim Teknolojileri
very good. thank you... <http://www.hdbilisimteknolojileri.com/>

Like · Reply · Dec 14, 2015 2:12am



Vitor Madureira · Faculdade de Engenharia da Universidade do Porto
Ok, and passing from view to controller?

Like · Reply · Dec 16, 2015 9:12am



Mohammad Rahat · Engineer at Student
So nice broher. Thank you very much

Like · Reply · Jan 31, 2016 1:47am

Load 4 more comments

Facebook Comments Plugin



File APIs for .NET

Aspose are the market leader of .NET APIs for file business formats – natively work with DOCX, XLSX, PPT, PDF, MSG, MPP, images formats and many more!

[ASK A QUESTION](#)

TRENDING UP

- 01 Multithreading In C# .Net
- 02 Best Practices For MVC
- 03 What Is Big Data?
- 04 Learn ASP.NET Core Using Angular 2
- 05 How To Use Joins, and Group By Clause In Entity Framework With LINQ C#
- 06 Confessions Of An Angry Programmer
- 07 Creating Web API With ASP.NET Core Using Visual Studio Code
- 08 How To Upload Records From Excel File to Database Using Stored Procedure In MVC
- 09 Future Of IoT, Machine Learning, And Artificial Intelligence
- 10 Project Server 2013 Migration From One Farm To Another

[View All](#) 

Follow @csharpcorner  96.3K followers

[Like Page](#)[Learn More](#)

Be the first of your friends to like this

[ASK A QUESTION](#)[Philadelphia](#)[New York](#)[London](#)[Delhi](#)

JOIN C# CORNER

A community of 2.3 million developers worldwide

[Sign Up](#)[Learn ASP.NET MVC](#) [Learn ASP.NET Core](#) [Learn Python](#) [Learn JavaScript](#) [Learn Xamarin](#)[Learn Oracle](#) [More...](#)[Home](#) [Events](#) [Consultants](#) [Jobs](#) [Career Advice](#) [Stories](#)



[About Us](#) [Contact Us](#) [Privacy Policy](#) [Terms](#)

[ASK A QUESTION](#)

[Report a Bug](#) [FAQ](#)

©2017 C# Corner. All contents are copyright of their authors.