

Learn MVC Project in 7 days – Day 1



Shivprasad koirala, **Marla Suresh**, 15 Nov 2017

In this article we learn MVC 5 step by step in 7 days – Day 1.



Download WebApplication1.zip

Contents

- Introduction
- What do we need for doing Asp.Net MVC ?
- ASP.NET vs MVC vs WebForms
- Why ASP.NET Web Forms?
- Problems with Asp.Net Web Forms
- What's the solution ?
- How Microsoft Asp.Net MVC tackles problems in Web Forms?
- Understand Controller in Asp.Net MVC?
- Understand Views in Asp.Net MVC
- Lab 1 – Demonstrating Controller with a simple hello world
- Q & A session around Lab 1
- Lab 2 – Demonstrating Views
- Q and A session around Lab 2
- What is for Day 2 ?
- Learn MVC in 2 days (16 hours)

Complete Series

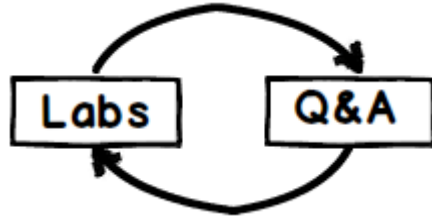
- [Day 1](#)
- [Day 2](#)
- [Day 3](#)
- [Day 4](#)
- [Day 5](#)
- [Day 6](#)
- [Day 7](#)
- [Bonus Day 1](#)
- [Bonus Day 2](#)

We are pleased to announce that this article is now available as hard copy book you can get the same from www.amazon.com and www.flipkart.com

Introduction

As the title promises “[Learn MVC step by step](#) in 7 days”, so this article will have 7 articles i.e. 1 article for each day. So start reading this tutorial series with a nice Monday and become a MVC guy till the end of the week.

Day 1 is kind of a warm up. In this first day we will understand Why Asp.Net MVC over Webforms ? , Issues with Webforms and we will do two Lab's one around controller and the around views.



After each one of these lab's we will run through a small Q and A session where we will discuss concepts of the lab. So the structure of this is article is Lab's and then Q and A.

In case for any Lab you have questions which are not answered in the Q and A please feel free to put the same in the comment box below. We will definitely answer them and if we find those question's recurring we will include the same in the article as well so that rest of the community benefit from the same.

So we just need your 7 day's and rest this article assures you become a ASP.NET MVC developer.



What do we need for doing Asp.Net MVC ?

We just need Visual Studio and the good news is that visual studio is completely free. You can download Visual studio community edition from <http://www.visualstudio.com/> , no expiry , no license issues and you do not need deep pockets.

ASP.NET vs MVC vs WebForms

"You are reading this article because you know ASP.NET and you want to upgrade yourself to MVC."

Sorry for the trouble, can you please read the above statement again and if you think it's right then this section is a must read for you.

Lot of ASP.NET developers who start MVC for the first time think that MVC is different new , fresh from ASP.NET. But the truth is ASP.NET is a framework for creating web application while MVC is a great architecture to organize and arrange our code in a better way. So rather than MVC you can say ASP.NET MVC.

Ok so if the new thing is ASP.NET MVC what is the old thing called as,it's "ASP.NET Webforms".

Let me correct your vocabulary:-

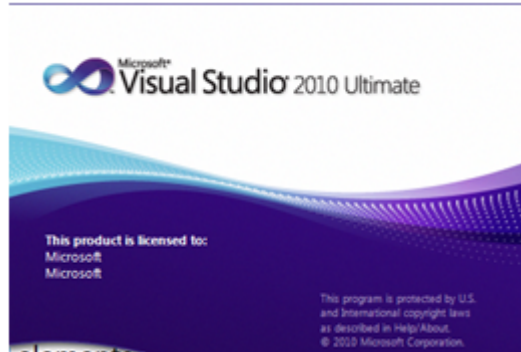
"You are reading this article because you know ASP.NET Webforms and you want to upgrade yourself to ASP.NET MVC."

So now that your vocabulary is corrected welcome to the world of ASP.NET MVC and let's start this tutorial.

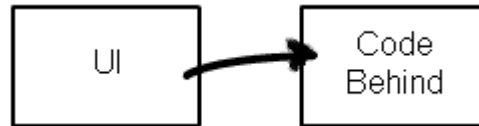
Why ASP.NET Web Forms?

ASP.NET Webforms has served and successfully delivered web application for past 12 years.Let us try to understand the secret of what made Webform's so popular and successful.

If you see the success of Microsoft programming languages right from the days of VB (visual basic) it is due to RAD(Rapid application development)and visual programming approach.Visual programming was so much preached and successful in Microsoft that literally they named their IDE's as "Visual studio".



By using visual studio ,developers where able to drag drop UI elements on a designer area and at the backend , visual studio generates C# or VB.NET code for those elements. These codes where termed as “Behind Code” or “Code Behind”. In this code behind Developers can go and write logic to manipulate the UI elements.



So the visual RAD architecture of Microsoft has two things one is the UI and the other is the code behind. So for ASP.NET Web forms you have ASPX and ASPX.CS ,for WPF you have XAML / XAML.CS and so on.

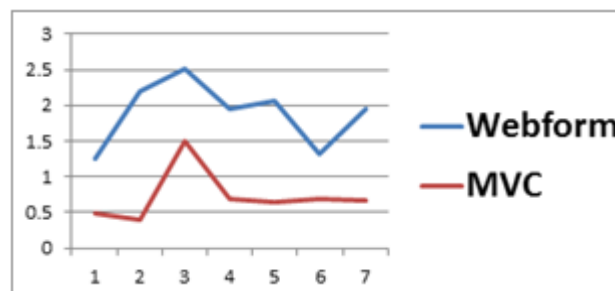
Problems with Asp.Net Web Forms

So when ASP.NET Webform was so successful, why Microsoft thought of creating ASP.NET MVC.The main problem with ASP.NET Webform is performance, performance and performance. In web application there are two aspects which define performance:-

1. Response time: - How fast the server responds to request?.
2. Bandwidth consumption: - How much data is sent ?.

Response time issues

Let us try to understand why response time is slower when it comes to ASP.NET Webforms. We did a small load testing experiment of Webform vs Asp.Net MVC and we found Asp.Net MVC to be twice faster.



Read more on how this test was done from [here](#)

Let us try to understand why ASP.NET MVC was better in performance in the above load test.Consider the below simple UI code and Code behind for that UI.

Assume the ASPX code has the below simple text box.

```
<asp:TextBox ID="TextBox1" runat="server">
```

In the code behind you have written some logic which manipulates the text box values and the back ground color.

```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox1.Text = "Make it simple";
    TextBox1.BackColor = Color.Aqua;
}
```

When you run the above program below is the HTML output.

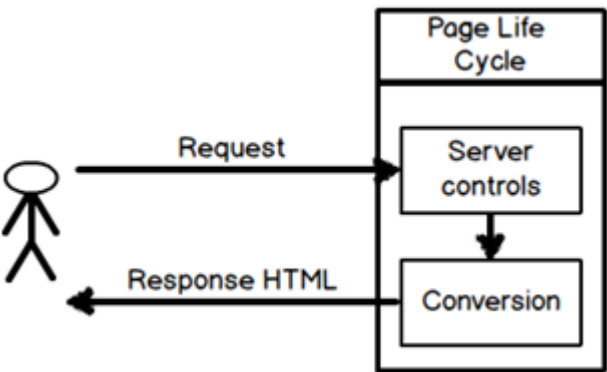
Make it simple

If you see the HTML output by doing view source it looks something as shown below.

```
<input name="TextBox1" type="text" value="Make it simple" id="TextBox1" style="background-color:Aqua;" />
```

Now stop reading for a moment, close your eyes and think. Try to get answers to the below questions:-

- 1. Is this a efficient way of generating HTML?. Do we really need to make those long server trips to get those simple HTML on the browser ?.
- 2. Can't the developer write HTML straight forward , Is it so tough?



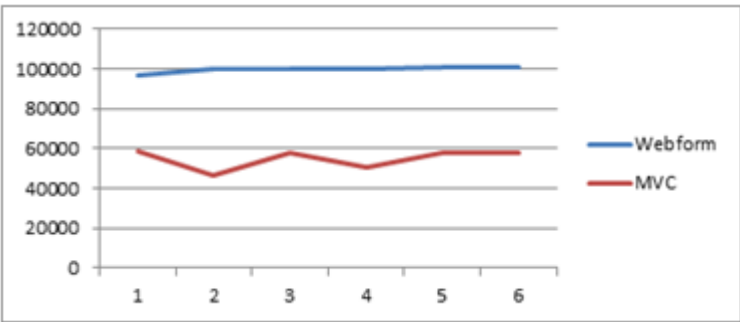
If you see for every request there is a conversion logic which runs and converts the server controls to HTML output.This conversion get's worse and heavy when we have grids, tree view controls etc where the HTML outputs are complicated HTML tables. Due to this unnecessary conversion the response time get affected.

Solution for this problem: - "GET RID of CODE BEHIND" ,fold your sleeves and work with pure HTML.

Bandwidth consumption

Viewstate has been a very dear and near friend of ASP.NET developers for past 10 years because it automatically saves states between post backs and reduces our development time. But this reduction in development time comes at a huge cost ,viewstate increases the page size considerably. In this [load test](#) we found viewstate increases the page size twice as compared to Asp.Net MVC.

Below is the plot of the content length emitted from Webform and Asp.Net MVC.



The size increase is because of extra bytes generated from viewstate , below is the snapshot of a viewstate. Lot of people can argue that viewstate can be disabled but then we all know how developers are , if there is option given they would definitely try that out.

```
<body>
<form method="post" action="WebForm1.aspx" id="form1">
<div class="aspNetHidden">
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="bfRayhg09cEK735FA/bNsy7ds1d0NoQ88njmU11+1twCvnpM660QlvR1h/Rx10EopzEz6XyqChdzCj5BHF9LRF6ZzD+SAH3XzF3vELxp9IhndKRZATU8Na3y0iaXgPDu0vRLu61xikXkQ6q817F7i
Glx9L8D60vzPj35c1PDB1XgCUg0ra7HdbuL3HT8QsL43Zf1EehAxet0VUtaeBuRHR1KoaactcnVd8c8U40VYjnn0zo90hdu11x7/63E0hian/B7XsP4jnr8zloQ0PCv7yQQssshVD1K6pG/Q17Lesv8NcPe2
cmQ0gnf0X0K7L7X5uizbzc+H4UyJekqB8C3X0E0u6XZ855UisqnpOILAB8mC2E/66mLQ1VP6UpQC5NDCHY8uL5C1Yh7Su9ywl701H23IQ5gE1tex004CmoD1Pvq+H4E1d20Tcf3xH84zR9aX1zqlm4
7kzq0pD1PHv709H+4K50qCkEj3Dg81z3or3Q1d9q1U0upUv0hAvcE5ut+P101d9q1d0HTHRndrbd5blm7e08myTndbHC06z08/3y70KvY9z8EplvH89yZm01d8tw1VPh1WtH2VndrsEArWfymk3w1xV7
r5YHnanvPL2Dn1Iay9b63PjnrHvNdl1DdfH5ySR1AG0u4yf8Hc4h6dCctSCJ1K2GvZnfmcHPe1f5z3bh08pd60GKJN+uQ0y3Y0trB488PYsash0vpgH44x77cK5ymLUIzqnrsxCh4HQ2pZUT1lyF
Q0uufT1cVchd46E6fV3QL5aIXKc6P3H21Z7yhaSVK0nssq/p61gLyaeQf346jrcvFDDP6y5XntucP8rR4E1NbHgrhplNdu0VhN7+/6jsE8Xds08Reve455my2oJYhKXF/s6v6tAcHc27Q87gV21/0u
P2h3o8906Aa31LURdsE8g579AmC9j04+svQ00EKA/Em4tA+1Ac+Huo0m7y3HT3vz5QpL8aefq34K73Xz7D5z5XK5ZC0ud8Kps5Fv311CRTq05kXVCK0uk9HMA7Ep7X0pkiap452AD0e2V73j0XQ+fur
h4QYC60vzrH986Den8Xgq9u4577737Tg8ing7QpPb8TVAw4h4F/7ADrvvq9K69DgP2vc4uAac75Znxfl8w2BqleaZ3CA3CIu+3xpFErrs1zc689V1Hb5sDfanXc0e0TClg10fuktaz5Bk879KX8tI
1x0Fnnv0A1Thad9966K0v4hVdV2tq/UmeCm8pCDeec4qht56t8n605U4P3HHRPw1nta8v0v8cPeh1n10v13gd100ePn/wb9t305Fw1F51A00g95Aja/ZodaJf+/nd1ZguLv46p0c8h80v208X1bori
m8Kxul1kwr19z2a0d8m30sJh+ze1CcsjFuccQ8806u7hG48KEdJdsdCnh57CPc9t1+b07Y0ecy81FYfK10vyyx09T35Fj0frj0iafz2xfgkZqdT11Q0bsnQdU+uYTDK00ZTP9Qcs/8s77XJ0YjC08si
yHq8A9uf90d0X3D0uKlqak+8K5svHdQ0aZQ80s28H0K+2VB8Z2PjE5V95ChkQ0hEIZX0/+YH09Kfus2905Irn1Lex8d5/B/zbfu82TwAkRyRmu3uW67n2Hu0Df8/VL5Se+29oq4zbb21V30jG1F4c
4g4Xa/O1VP0xk1d8vC1P8bdKcXsfHjYX35Y0Lrfs0o2LAuyU0jy+B+hvhiwbs1gRck0Aab8YTdr1p+63A/9etzhomEujR/dh5Kueg4ta7X+q6KDuQ8K5gp0K1u0Y9AZV25FP9/27Q0d5GAsqLD5mJ
3611c4QpGtZ1ztya+Q2m8c6P0HmE9r755z2m3sLLV0Z0S+rj2ef/aAYCr66c5UFRF93zoara3PQ200c4N8U/zfWcJ7z03haD7Hr5vLC0nc4m1Cn8yanR7Pg8Xtq+4HakX17VRyW154E2h941j7L37A140rC
jKZnR1f1dQ0d80kAp681q2t9IIasw20qCk0e6XtfcTq5EF1JEHRUXVsp1jxkfoto70/KnJ1o1NUSQ5206ZdTckkBePve2Heg1gtR4c4k7AF1yyH2f5srFSU71I0Vgq0PE1dc1u68cITQ1F2TK5Y940Q0R
Eko8Nvhd0z3D08tegdsR4EtmclvDv4Ytb3758xy00Q+AS3Pb6/P9qKaunrveA13xUpK1oX1Idq1/+2y51281k+0u0j+X2sAht0Y7x8KSP0BuaM5EHF1RzberHfGJcatu3140018e0e+/J/L7t9/zg0Hf.
tLx0+dtbu7lk770oTKF429AX/GHdTK6D9sXcTuaXsSroqD7PB7nIVBj69xydx+eH0LUYtT02q56V550F8H23Q1q0qKTLH6GrZj+anu7jshgZ6CyhQ9213LYB157Xf091b7TgdGr5uePz9Hyq9CqUj
I6iaH4Z0hN45xun3Fst0X/pj4eJ6P1H2yk03PqT3t1u0Luh5S7/q4U088Xqu2Fzr+RLn59E26P1D0c4d8LdhtsYc+q0kolP4Zbkznz8x5Zo6zFcK8R1HsRsk90wG1Dg0glebrTtRaL126bu7fGnkF1g7qp8
CPH61g3s1x850/c2H0Vcfc7X4RQ7P1Pks0ty+0m/q4CbLDGtXt1IkTISHC0TH4u0v4HJmnsE3H/v0uRD8zPgtR6b2VIQ/1Czsp+BP0dg257j7p7HR1H0sssc2RL0HtE157Psu0Y4eL9X0I2HmJhao32P0
KIS/H0Coby8f3dxs1H7xngD09v0jcs3EE09HglakUG81JChIP7A0L7cb/s5n/77+v2ahpdgaH1e0pp/bf4P/PF65+mC2bdfCw6FvQECuILFXtnRen0HzRQX511IntkOGHf5THt7eH8dHT1t8rk/bjHrsILU
VHndIeyx1pt2opmeflr3SR1HEKcgkI6bk0L3u7jQZ0W0Q1c4jEV26m12DVyCPUCH1BaYag8H08c28G/R4EH+BcwQmc5jB6dQzqnD5d0+t3kURncISu0R2Vg7UhiES/uUPzhiwF0K0uW/+3gVC0R8Puz8
tu4NhtDoyymvuaB1Urf8t/VzRw0byq28hAK5638AeuQ3Au7hyrdRncU8H2g+cqPwa0yLOArVufZ/Q7Uz7heoK1b3GomDsw8Rv4gIAousFFD4Xs2F4yoxtdfC8BIVxcUUD05qnuUj5zSK/z7oN0KCI6+00q
v08bsn85M10u78HyVHTfyf1sk0Jd3uRzIS60pKottVfH6K7ID6FQ7nbZnjah7VplwadFP7q2PaQm2VPA7nNab9zkelHRvR55/zHEIdeBmmZeRR0vfpGZt91bd1wzfiAFrXfPaxjypluecsdqISkdZaa
/nxtNF+74C0Hb1Pxs+E1ycd3XulBur1UKxRPncd19gDnHwgdXexF092Kz5D0PC7Rw111huos7FDn0dp46/Hd/n1mFh1dLPQLa7t1gVB5u5nkgf8Cnld7VPQot7ztH0HsxEx24HQ01+4HahB5RYV+C4gd
xmZf6s0C4A0qJ6dU6jZ5s0mp6wUj10RKK0P2n9vjsyefURcpVUE0B+fvKc+qzVQd8ht60g+fv5pYRykADN19Kde4xtL0ELEUkg0BQBU0m92N7meP9TwpfYx5tdVp8v9C8g3Tzc14Pdadu0U8Eh/df19ag
xp1FHAAC11vg1a77YHBSnTXuZtH0b/7H57z1k3fd7Q6x93T/wY38fuCmKXGuacTCKZhsuQ5axCAJ+RaaT8hKU49S9R6trnxpyH4JRxtdgOm4P8oTxHrk8qkhVklal8lrvnEzJn3MY9FR09pK1zI
X68880e82Luf0eTxCcc750vvDv40KLL08DHzc+cf+0R5SLVY1V13T0TFrvBe8e5Vv630E3oEsF1s2X9tE1I1R1w13aaoLs2ErX1vXT305Zcda8vFA3Kd+hFR0T+7anV9uHCE3HTKBFd5c3Hf6d0F/nA8
```

Solution for this problem: - “GET RID of SERVER CONTROLS”.

Note: -The rest of the three points down below are brownly issues which have cropped up due to presence of code behind and server controls. But the main thing is always performance.

HTML customization

Now because we are saviors of the code behind and ASP.NET web server controls, we have “NO IDEA” what kind of HTML can come out and how efficient they are. For example see the below ASPX code, can you guess what kind of HTML it will generate.

```
<asp:Label ID="Label1" runat="server" Text="I am label">
<asp:Literal ID="Literal1" runat="server" Text="I am a literal">
<asp:Panel ID="Panel1" runat="server">I am a panel
```

Will Label generate DIV tag or SPAN tag ?. If you run the above code below the respective generated HTML. Label generates a SPAN , Literal generates simple text , Panel generates DIV tag and so on.

```
<span id="Label1">I am label</span>
I am a literal
```

I am a panel

So rather than generating HTML using server controls how about writing HTML directly and taking complete control of HTML.

So the solution for this problem is “DO NOT USE SERVER CONTROLS” and work with direct HTML.

The other great benefit of working directly with HTML is that your web designers can work very closely with the developer team. They can take the HTML code put in their favorite designer tool like dream weaver , front page etc and design independently . If we have server controls these designer tools do not identify them easily.

Reusability of code behind class

If you watch any professional ASP.NET Webform project you will notice that code behind class is where you have huge amount of code and the code is really complicated.Now this code behind page class inherits from “System.Web.UI.Page” class. This class is not a normal class which can be reused and instantiated anywhere. In other words you can never do something as shown below for a Webform class:-

```
WebForm1 obj = new WebForm1();
obj.Button1_Click();
```

Because the “WebForm” class cannot instantiate with out “request” and “response” object. If you have ever seen the “ButtonClick” events of “WebForm” they are as shown in the code below. From the code you can know how difficult it is to instantiate the same.

```
protected void Button1_Click(object sender, EventArgs e)
{
// The logic which you want to reuse and invoke
}
```

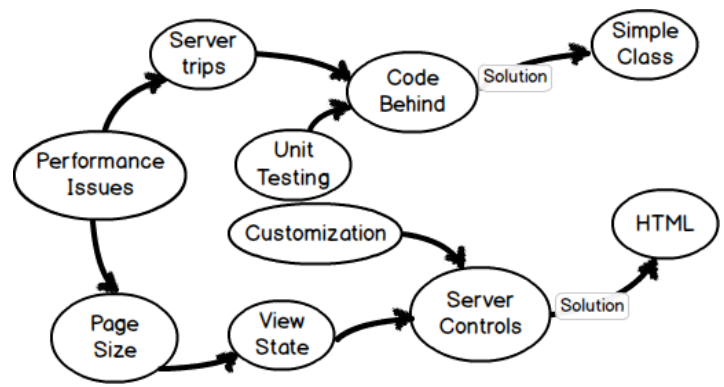
Solution for this problem: - “GET RID of SERVER CONTROLS and CODE BEHIND”.

Unit Testing

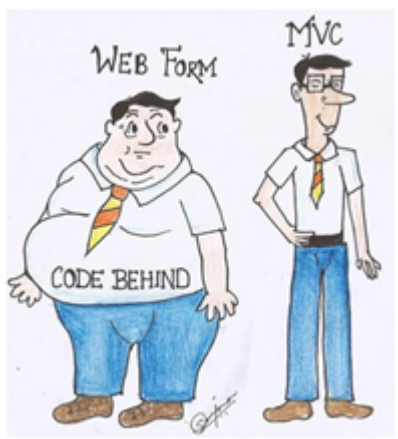
As said in the previous section you cannot instantiate behind code straight forward it's very difficult to do unit testing or I will say automation testing on the code behind. Someone has to manually run the application and do the testing.

What's the solution ?

If we read the four issues mentioned in the previous section with ASP.NET Webforms the main culprit are two people "Code Behind" and "Server controls". Below is root cause diagram I have drawn. In this I started with problems , what is the cause for it and the solution for the same. The complete diagram zeroes on two things "Code Behind" and "Server controls".



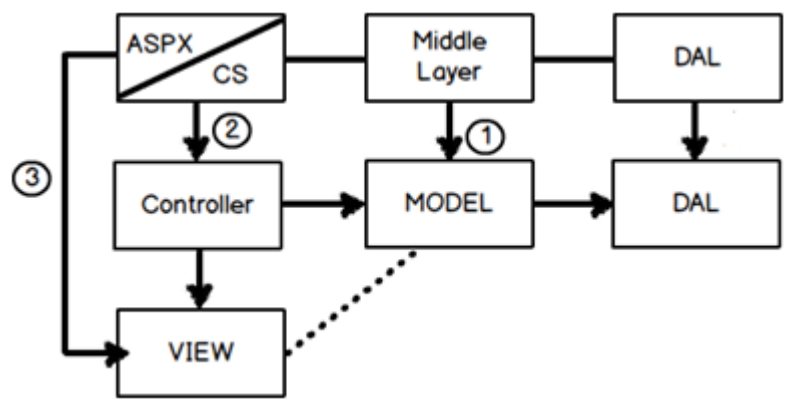
The solution is we need to move the code behind to a separate simple class library and get rid of ASP.NET Server controls and write simple HTML. In short the solution should look something as shown in the below image.



How Microsoft Asp.Net MVC tackles problems in Web Forms?

As said the code behind and server controls are the root cause problem. So if you look at the current WebForm architecture which developers are using it's mostly 3 layer architecture. This three layer architecture comprises of UI which has ASPX and the CS code behind.

This UI talk's with .NET classes which you can term as middle layer , business logic etc and the middle layer talks with data access layer.



So Asp.Net MVC comprises of three sections Model , View and Controller. The code behind logic goes in to the controller. View is your ASPX i.e. pure HTML and your Model is your middle layer. You can see in the above diagram how those layers fit in.

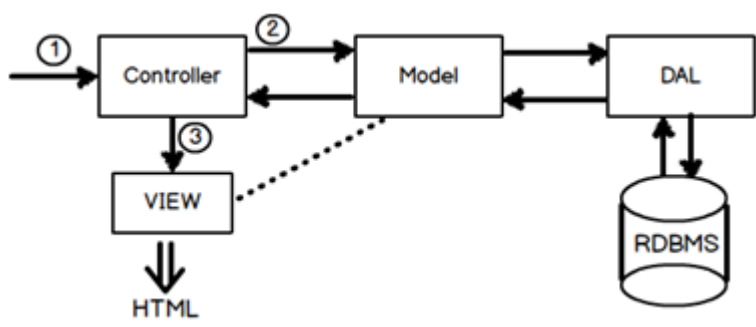
So if you see there are two major changes VIEW becoming simple HTML and code behind moving to simple .NET classes termed as controller.

In Asp.Net MVC request flow in general moves as follows:-

Step 1:- The first hit comes to the controller.

Step 2:- Depending on the action controller creates the object of the model. Model in turn calls the data access layer which fetches data in the model.

Step 3:- This data filled model is then passed to the view for display purpose.



Now that we have understood the different components of Asp.Net MVC let's go in depth in to each one of these components , let us start doing some lab's. Let us first start with controllers as they are the most important and central part of the MVC architecture.

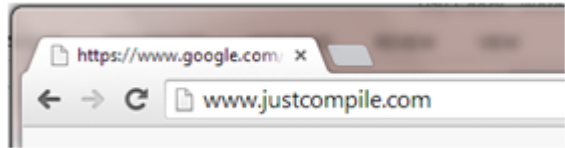
Understand Controller in Asp.Net MVC?

In order to understand Controller first we need to understand this term User interaction logic.

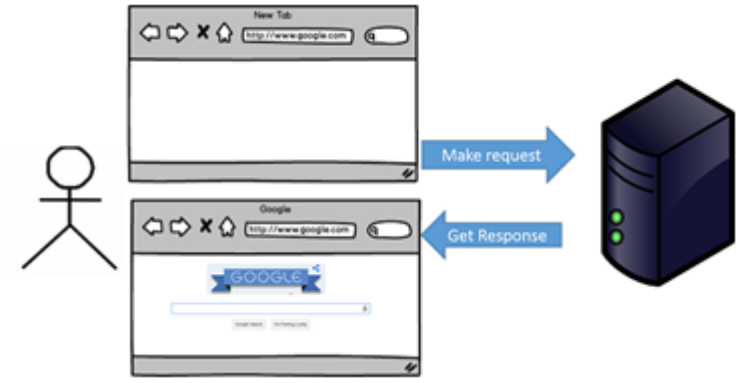
What is User Interaction logic??

Scenario 1

Did you ever gave a thought what happen's, when end user hits a URL on a browser.



Browser sends request to server and server sends a response.



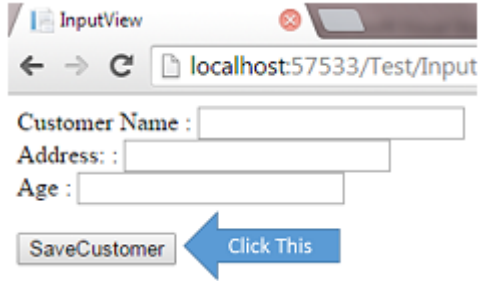
By means of such request, client is trying to interact with server. Server is able to respond back because some logic is written at server end to fulfil this request.

Some logic??, So what exactly can be this logic ?.

Logic which will handle the user requests and user's interaction with server. In short User Interaction Logic

Scenario 2

It also possible that response sent by Server is an HTML response. HTML response which can consist of couple of input controls and a submit button.



What will happen when “Save Customer” button is clicked?

If your answer is “Some event handler will handle the button click”, then sorry :-(.

“In reality in web programming there is no concept of event’s. In case of Asp.net Web Forms Microsoft wrote some code on behalf of us and brought us the feeling of event driven programming. It’s just an abstraction or the right word would illusion.”

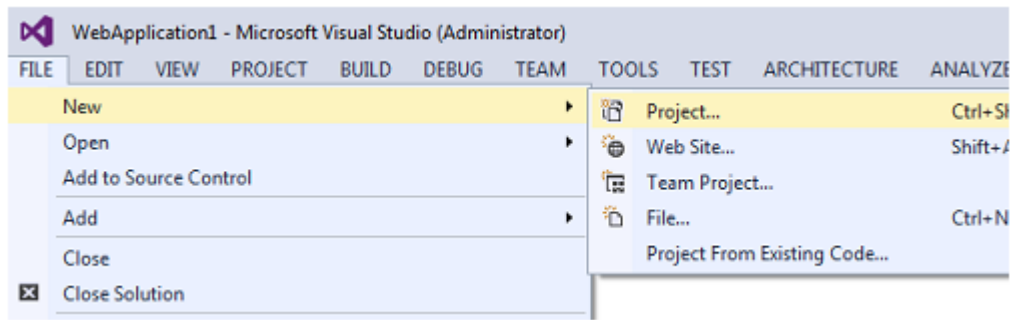
When button is clicked a simple HTTP request is sent to the server. This time the difference is, values in the “Customer Name”, “Address” and “Age” will be sent along with request. (In technical terms “values are posted to the server”). Ultimately, if it’s a request then there must be a logic written in the server so that server can send back the response. In short there must be some user interaction logic written on the server.

In Asp.Net MVC, the last letter C that is Controller is the one who will handle the user interaction Logic.

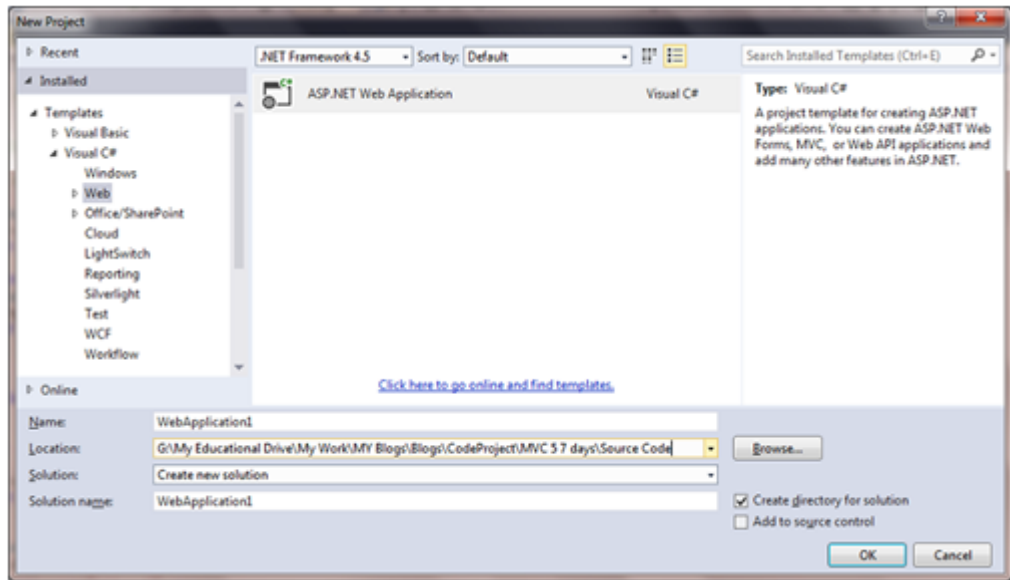
Lab 1 – Demonstrating Controller with a simple MVC hello world

Step 1 – Create Asp.Net MVC 5 Project

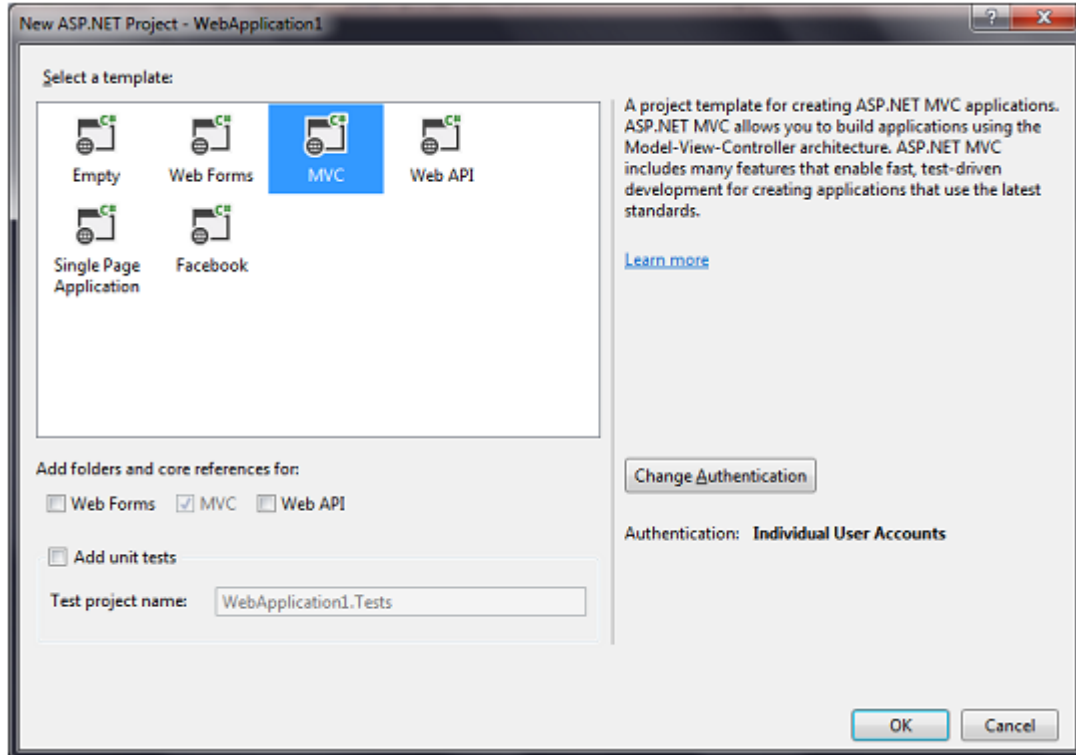
Step 1.1 Open Visual studio 2013(or higher). Click on File>>New>>Project.



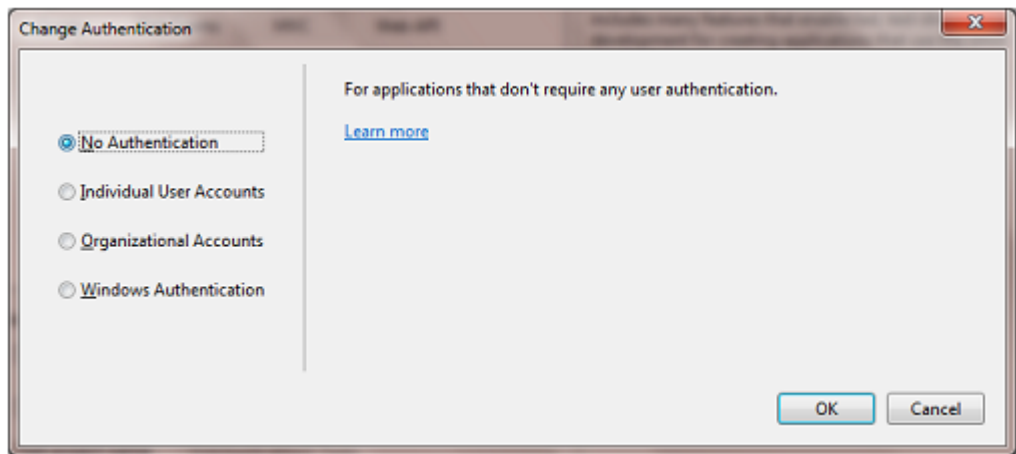
Step 1.2 Select Web Application. Put Name. Put Location and say ok.



Step 1.3 Select MVC template



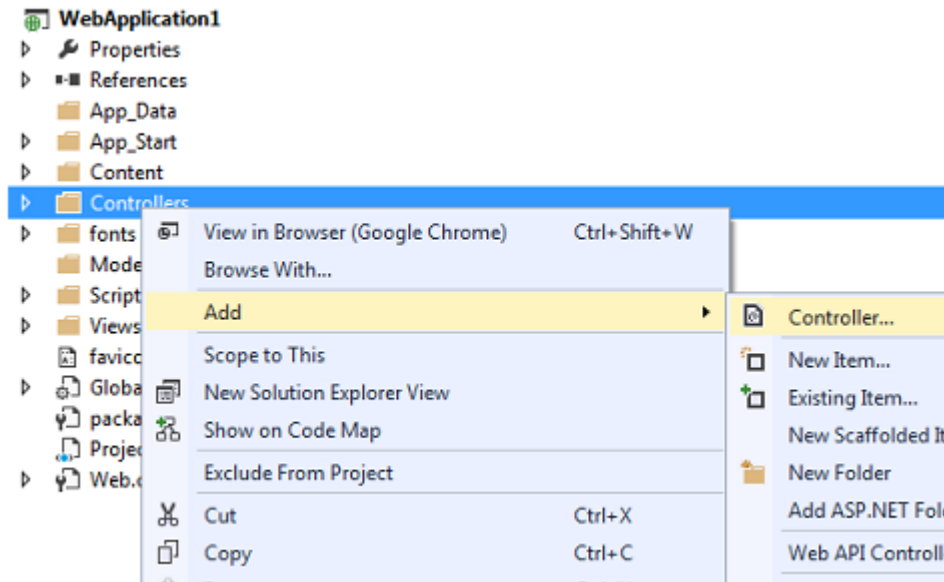
Step 1.4 Click on Change Authentication. Select "No Authentication" from "Change Authentication" dialog box and click ok.



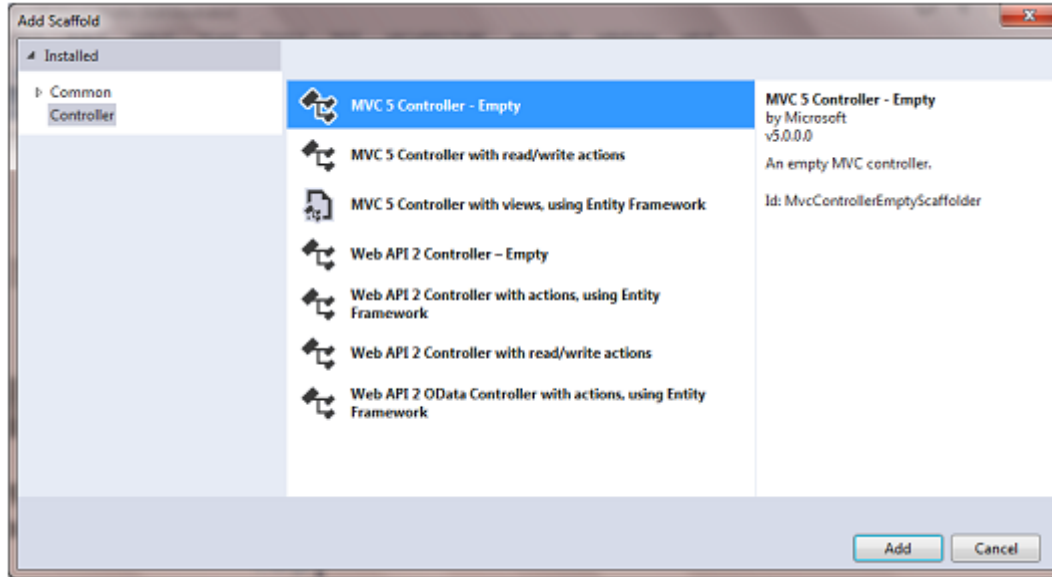
Step 1.5. Click ok.

Step 2 – Create Controller

Step 2.1. In the solution explorer, right click the controller folder and select Add>>Controller

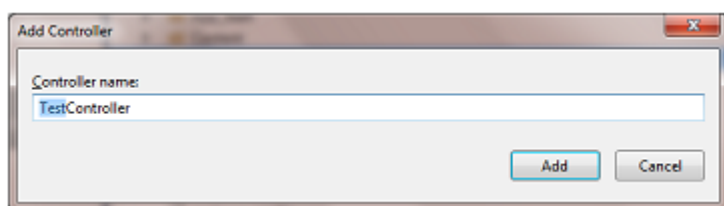


Step 2.2. Select "MVC 5 Controller – Empty" and click Add



Step 2.3. Put controller name as "TestController" and click Add.

One very important point to note at this step is do not delete the word controller. For now you can think it's like a reserved keyword.



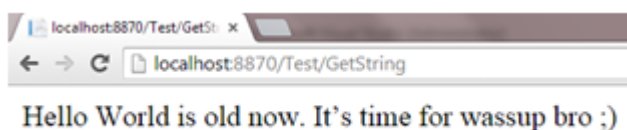
Step 3. Create Action Method

Open newly created TestController class. You will find a method inside it called "Index". Remove that method and add new public method called "GetString" as follows.

```
public class TestController : Controller
{
    public string GetString()
    {
        return "Hello World is old now. It's time for wassup bro ;)";
    }
}
```

Step 4. Execute and Test

Press F5. In the address bar put "ControllerName/ActionName" as follows. Please note do not type the word "Controller" just type "Test".



Q & A session around Lab 1

What is the relationship between TestController and Test?

TestController is the class name whereas Test is the controller name. Please note, when you type the controller name on the URL it should be without the word controller.

Asp.Net MVC follows Convention based approach. It strictly look's into the conventions we used.

In Asp.Net MVC two things are very important.

1. How we name something?
2. Where we keep something?

What is Action Method?

Action method is simply a public method inside controller which accepts user's request and returns some response. In above example, action method "GetString" is returning a string response type.

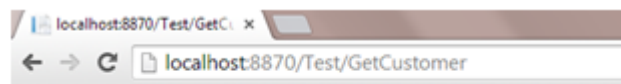
Note: In Asp.Net Web Forms default return response is always HTML. In case we want to return something other than HTML (in asp.net Web Forms), we create HTTP handlers, override content type , do response.end etc. It's not an easy task. In Asp.net MVC it's very easy. If return type is 'string' you can just return string :-), you do not need to send complete HTML.

What will happen if we try to return an object from an action method?

Look at the following code block.

```
namespace WebApplication1.Controllers
{
    public class Customer
    {
        public string CustomerName { get; set; }
        public string Address { get; set; }
    }
    public class TestController : Controller
    {
        public Customer GetCustomer()
        {
            Customer c = new Customer();
            c.CustomerName = "Customer 1";
            c.Address = "Address1";
            return c;
        }
    }
}
```

Output of above action method will look as shown below.



WebApplication1.Controllers.Customer

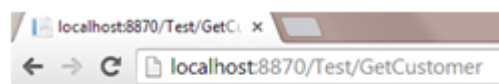
When return type is some object like 'customer', it will return 'ToString()' implementation of that object. By default 'ToString()' method returns fully qualified name of the class which is "NameSpace.ClassName";

What if you want to get values of properties in above example?

Simply override "ToString" method of class as follows.

```
public override string ToString()
{
    return this.CustomerName+"|"+this.Address;
}
```

Press F5. Output will be as follows.



Customer 1|Address1

Is it must to decorate action methods with public access modifier?

Yes, every public method will become action methods automatically.

What about non-public methods?

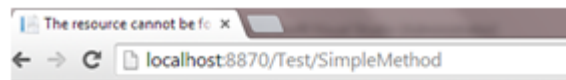
They are simply methods of a class and not available publicly . In simple words these methods can not be invoked from the web.

What if we want a method to be public but not action method?

Simply decorate it with NonAction attribute as follows.

```
[NonAction]
public string SimpleMethod()
{
    return "Hi, I am not action method";
}
```

When we try to make request to above action method we will get following response.



Server Error in '/' Application.

The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its: URL and make sure that it is spelled correctly.

Requested URL: /Test/SimpleMethod

Understand Views in Asp.Net MVC

As we discussed earlier controller will handle the user's requests and send the response. Most commonly the response is HTML as browser understands that format much better. HTML with some images, texts, Input controls etc. Normally in technical world layer defining the user interface design is termed as UI layer and in Asp.Net MVC it is termed as View.

Lab 2 – Demonstrating Views

In the first lab we created a simple MVC application with just controller and simple string return type. Let us go add view part to the MVC application.

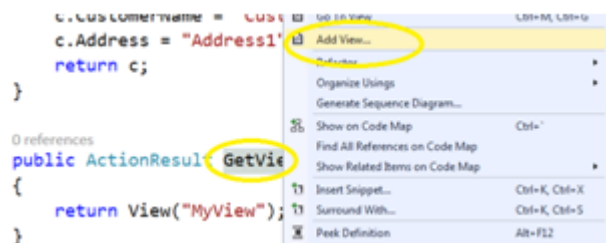
Step 1 – Create new action method

Add a new action method inside TestController as follows.

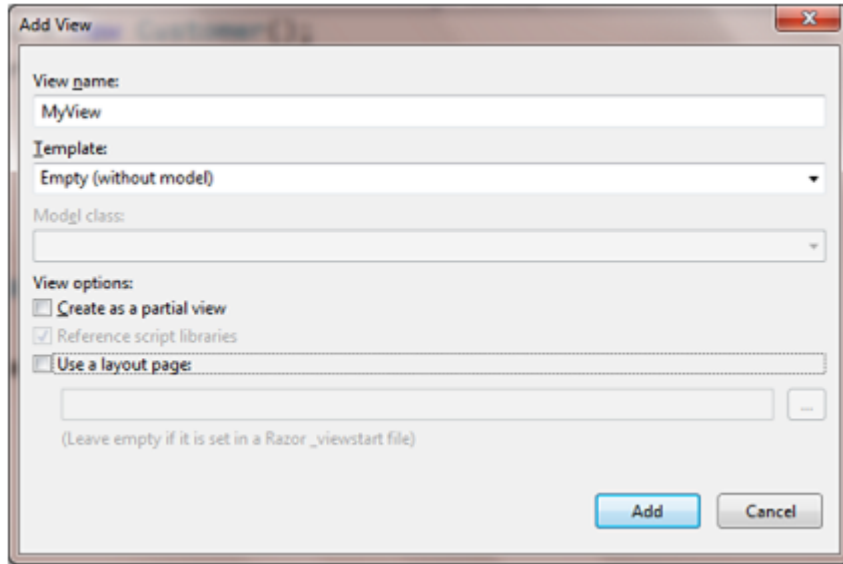
```
public ActionResult GetView()
{
    return View("MyView");
}
```

Step 2 – Create View

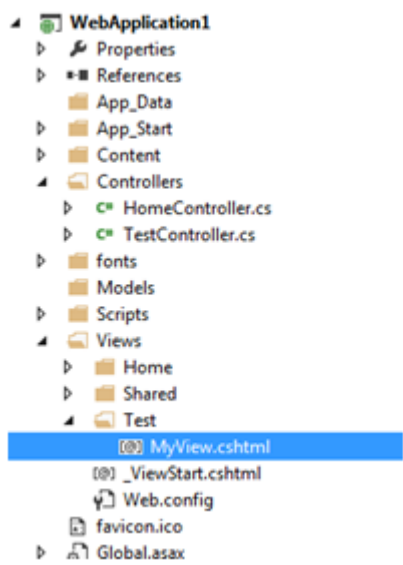
Step 2.1. Right click the above action method and select "Add View".



Step 2.2. In the "Add View" dialog box put view name as "MyView", uncheck "use a layout" checkbox and click "Add".



It will add a new view inside "Views/Test" folder in solution explored



Step 3 – Add contents to View

Open MyView.cshtml file and add contents as follows.

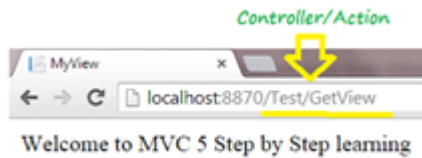
```
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title>MyView</title>
</head>
<body>
```

Welcome to MVC 5 Step by Step learning

</body> </html>

Step 3. Test and Execute

Press F5 and execute the application.



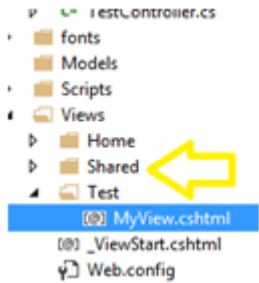
Why View is placed inside Test Folder?

In ASP.net MVC, Views associated with the particular controller is placed inside a special folder. This special folder will be named as “ControllerName” and placed inside Views folder (located in the root folder). For every controller only those views will be available which are located inside its own folder.

For example: All the views related to Test controller will be placed inside “~/Views/Test” and Test controller can access only those views which are inside Test folder.

Can’t we reuse some of the views across multiple controllers?

Yes, we can. For that we will keep those files inside a special folder called “Shared”.



Views located inside this Shared folder will be available to all the controllers.

Is it possible that one action method is referencing more than one views?

Yes. Look at the following code.

```
public ActionResult GetView()
{
    if(Some_Condition_Is_Matching)
    {
        return View("MyView");
    }
    else
    {
        return View("YourView");
    }
}
```

Note: In Asp.Net MVC views and controllers are not tightly coupled. One action method can refer more than one view and one view can be referred by more than one action method (by keeping them in Shared folder). It provides better reusability

What is the purpose of View function?

Creates ViewResult object which renders View to the response.

- ViewResult internally creates the object of ViewPageActivator
- ViewResult choose the correct ViewEngine and passes ViewPageActivator object as argument to ViewEngine’s constructor.
- ViewEngine create the object of View class
- ViewResult invoke the RenderView method of View.

Note: We have separate topic in the series disucssing Asp.Net MVC life cycle in detail.

What is the relation between ActionResult and ViewResult?

ActionResult is the abstract class whereas ViewResult is the multi level child of ActionResult. Multilevel because, ViewResult is the child of ViewResultBase and ViewResultBase is the child of ActionResult.

If we want to return ViewResult why ActionResult is the ViewResult?

To achieve polymorphism. Look at the following example.

```
public ActionResult GetView()
{
```

```
if(Some_Condition_Is_Matching)
{
    return View("MyView");
}
else
{
    return Content("Hi Welcome");
}
```

In the above example, when some condition is matching we are returning we are invoking "View" function which will return ViewResult whereas in some other condition we are invoking "Content" function which is returning Content Result.

What is ContentResult?

ViewResult represents a complete HTML response whereas ContentResult represents a scalar text response. It's just like returning pure string. Difference is ContentResult is aActionResult wrapper around string result. ContentResult is also the child of ActionResult.

Is it possible to invoke View function without Parameter?

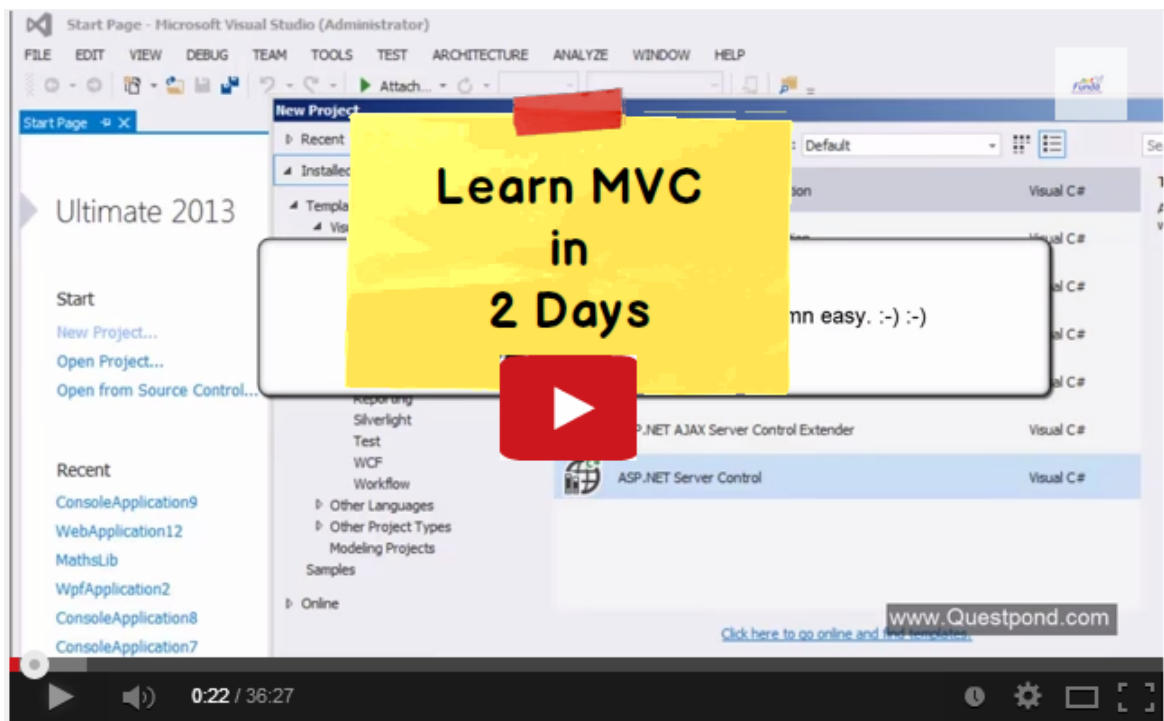
Yes, then it will find the view with name "CurrentActionName".

What is for Day 2 ?

In day 2 we will talk about Models , Validation , JQuery and Json. So keep rocking , keep learning.

Learn MVC in 2 days (16 hours)

We both authors have started an initiative Learn MVC in 2 days i.e. Flat 16 hours. Below is the video for the same. If you are Mumbai you can always connect us at <http://stepbystepschools.net/>



License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)

About the Authors



Shivprasad koirala



Architect <https://www.questpond.com>

India 

Do not forget to watch my Learn step by step video series.

[Learn MVC 5 step by step in 16 hours](#)

[Learn MVC Core step by step](#)

[Learn Angular 2/4 for beginners \(step by step\)](#)

[Learn AngularJS 1.x Step by Step](#)

[Learn Azure Step by Step](#)

[Learn Xamarin Mobile Programming Step by Step](#)

[Learn Design Pattern in 8 hours](#)

[Learn C# in 100 hours series](#)

[Learn SQL Server in 16 hours series](#)

[Learn Javascript in 2 hours series](#)

[Learn MSBI in 32 hours](#)

[Learn SharePoint Step by Step in 8 hours](#)

[Learn TypeScript in 45 minutes](#)



Marla Sukesh



Instructor / Trainer Train IT

India 

Learning is fun but teaching is awesome.

Code re-usability is my passion ,Teaching and learning is my hobby, Becoming an successful entrepreneur is my goal.

By profession I am a Corporate Trainer.


I do trainings on WCF, MVC, Business Intelligence, Design Patterns, HTML 5, jQuery, JSON and many more Microsoft and non-Microsoft technologies.

Find my profile [here](#)

My sites

- justcompile.com
- www.sukesh-marla.com

Comments and Discussions

 **256 messages** have been posted for this article Visit <https://www.codeproject.com/Articles/866143/Learn-MVC-Project-in-days-Day> to post and view comments on this article, or click [here](#) to get a print view with messages.