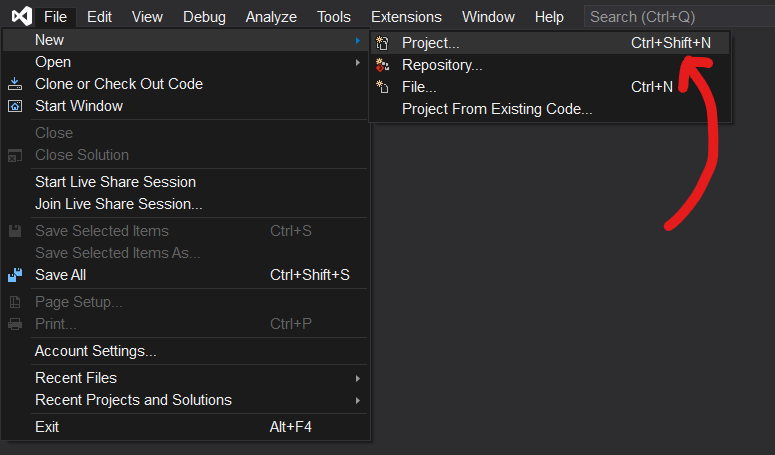
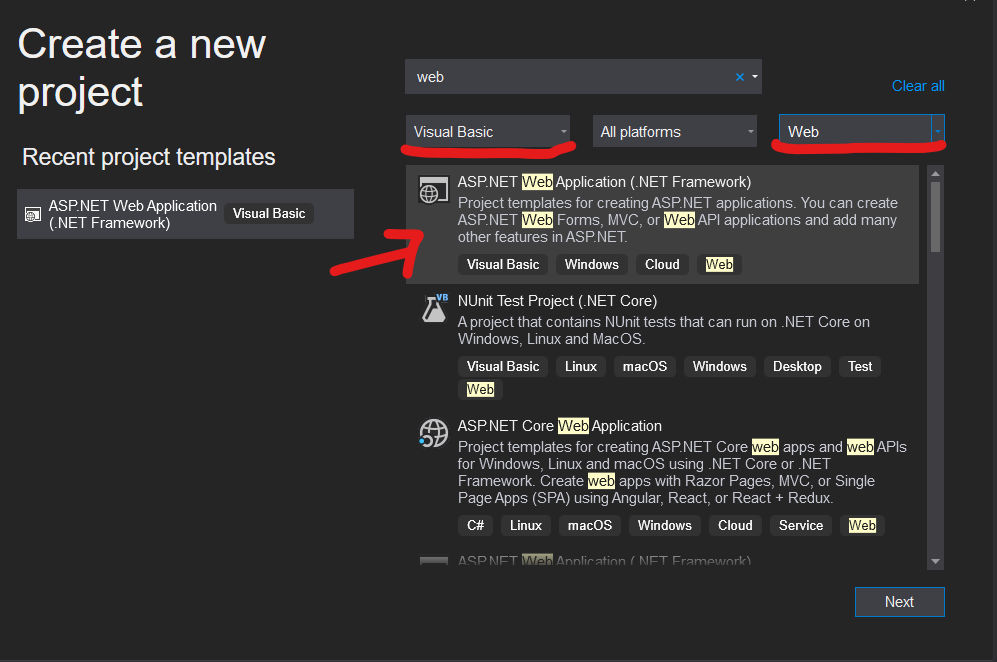
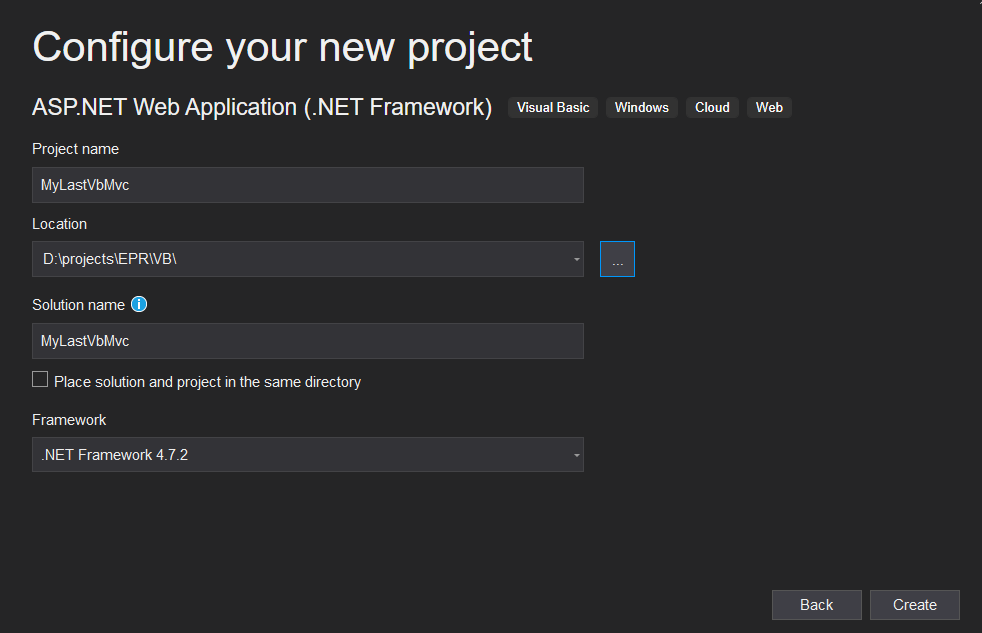
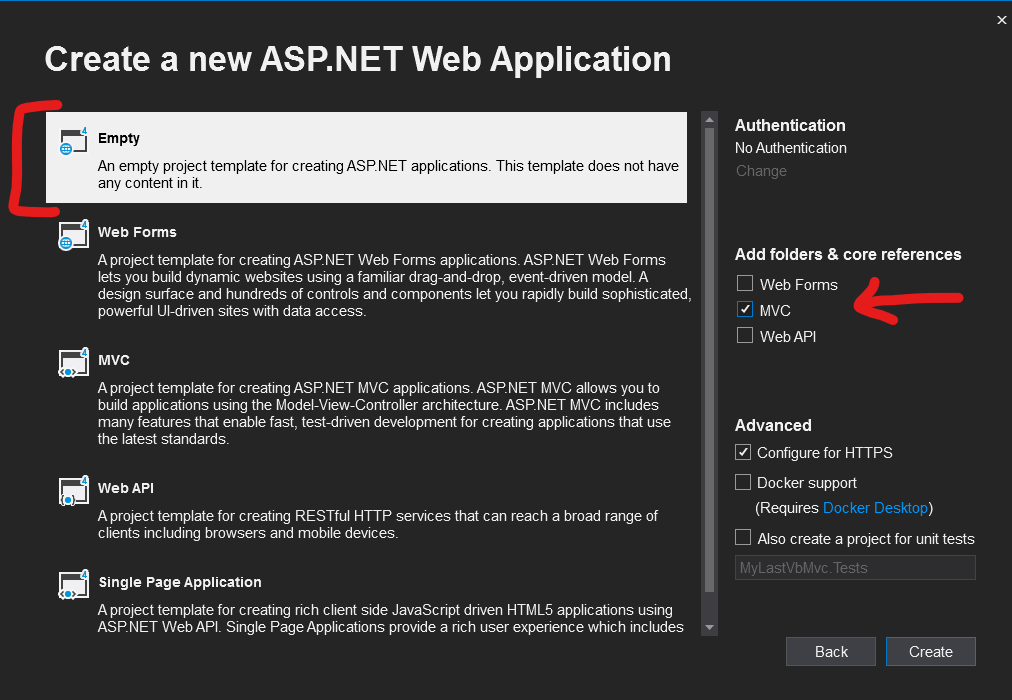
## Build My Last MVC vb Project

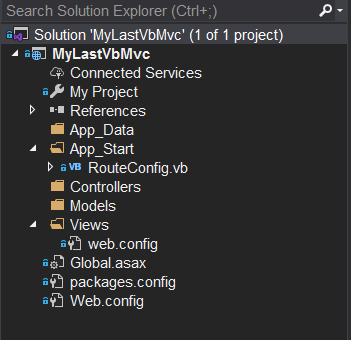
File > New > Project

Find “web” > Visual Basic & Web > APS.NET Web Application > **Next**

Project Name, Location, Solution Name > **Create**

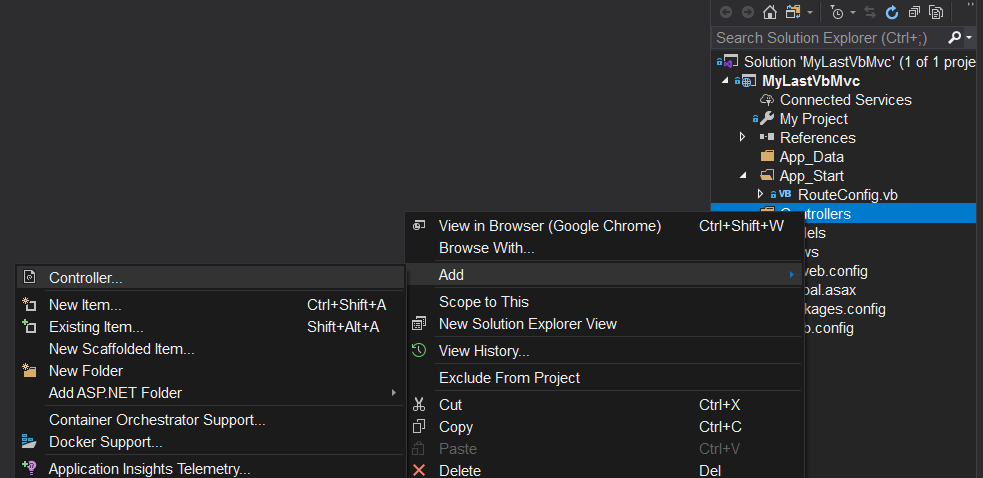
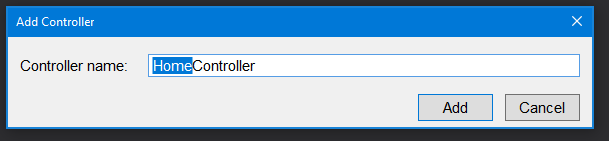
Mark Empty > check MVC > **Create**



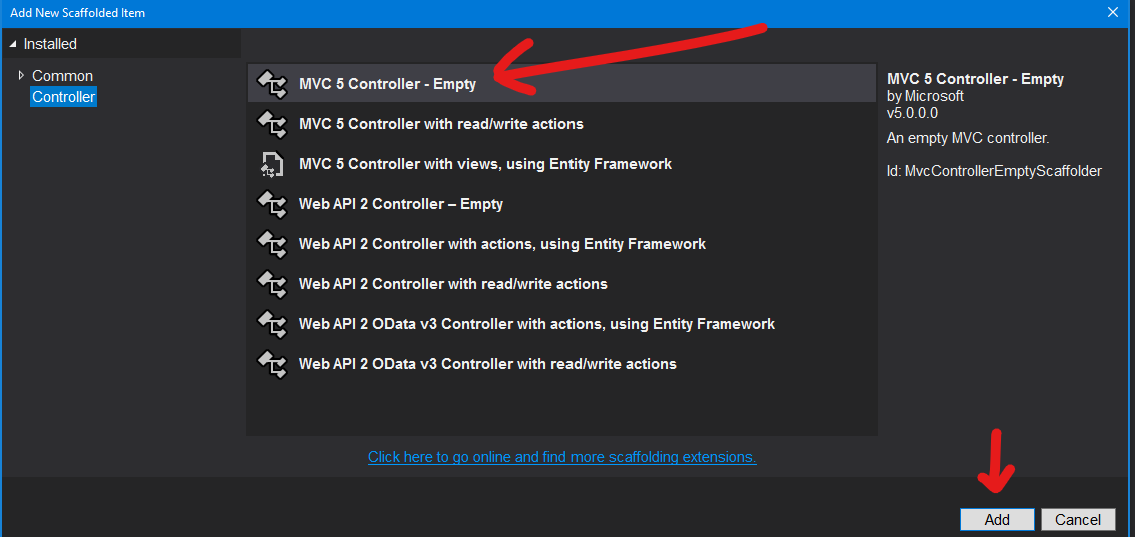
Our Empty Solution:

Now, Let’s make it a MVC Project!!

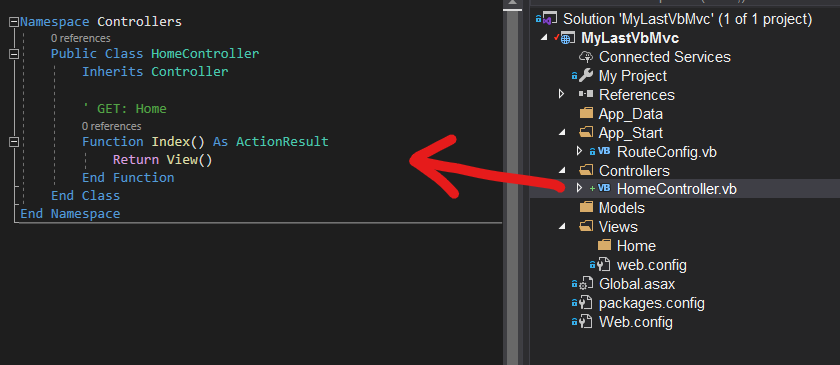
First step is Controller:

Right click on Controllers > Add > Controller

Mark Empty



Give a Name to the controller but keep the suffix Controller:

Congratulation! We just create our First Controller!

We can see that our controller is a class with special name **Home**Controller and Inherits Controller.

By default we get **Index Action.**

Public Class HomeController

Inherits Controller

' GET: Home

Function Index() As ActionResult

Return View()

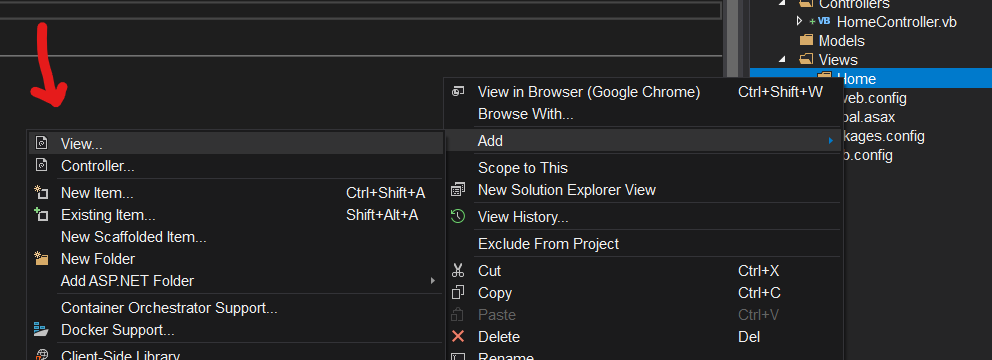
End Function

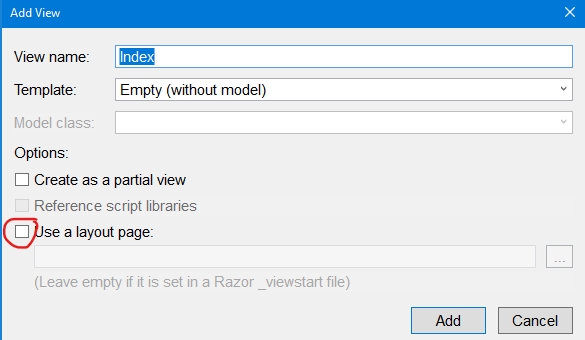
End Class

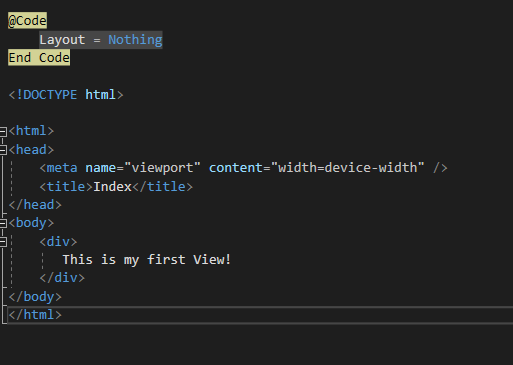
**Action –** The Controller’s method that we want to expose to the client.

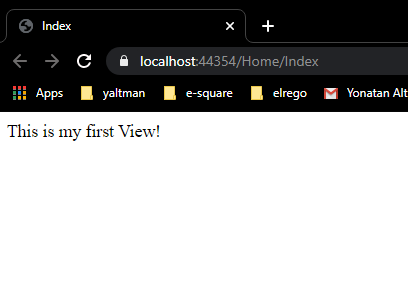
Return View() have 7+ overloads. If its empty, it will try to find a View with the same name as our action under the Views\<controller\_name> in our case is “Home”.

Return View("Home") will return the view with the specific name we choose.

Right click on Views\Home > Add > View

View Name : Index > unchecked User a layout page > Click **Add**

The new file is Index.vbhtml witch is combination of VB & Html:

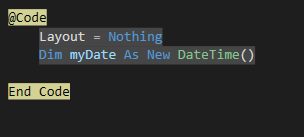
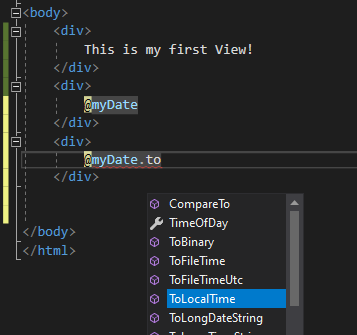
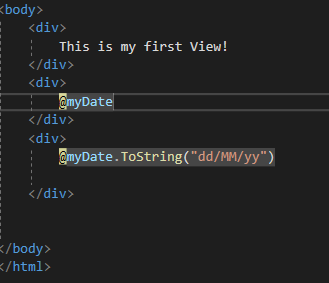
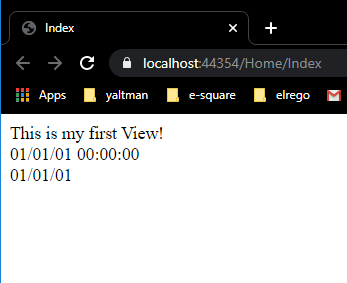
Now Press Ctrl + F5 for build & host our app to an iis express server. Change the Html and see the changes in the browser:

# Passing Data From Controller To View:

1. Razor
2. ViewBag
3. ViewData
4. TempData

## 1. Razor

The razor used by the @ sign:

 Inside the @Code – End Code its like writing code in a .vb file, and the we can access the data with @ and the name of the data:

## 2. ViewBag

ViewBag can be useful when you want to transfer temporary data from the controller to the view. The viewBag is a [dynamic](https://www.tutorialsteacher.com/csharp/csharp-dynamic-type) type property of ControllerBase class which is the base class of all the controllers.

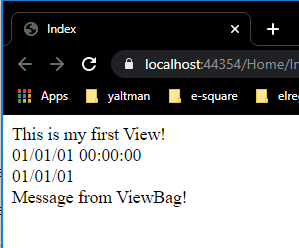
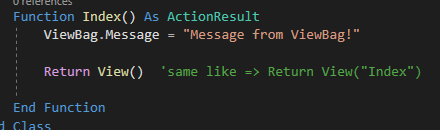
**In the Action Index:**

ViewBag.Message = "Message from ViewBag!"

**In the View:**  
<div>

@ViewBag.Message

</div>



## 3. ViewData

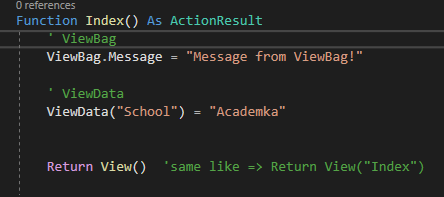
ViewData is a dictionary which can contain key-value pairs where each key must be string.

**In the Action Index:**

ViewData("School") = "Academka"

**In the View:**<div>

The best school is: @ViewData("school")

</div>

