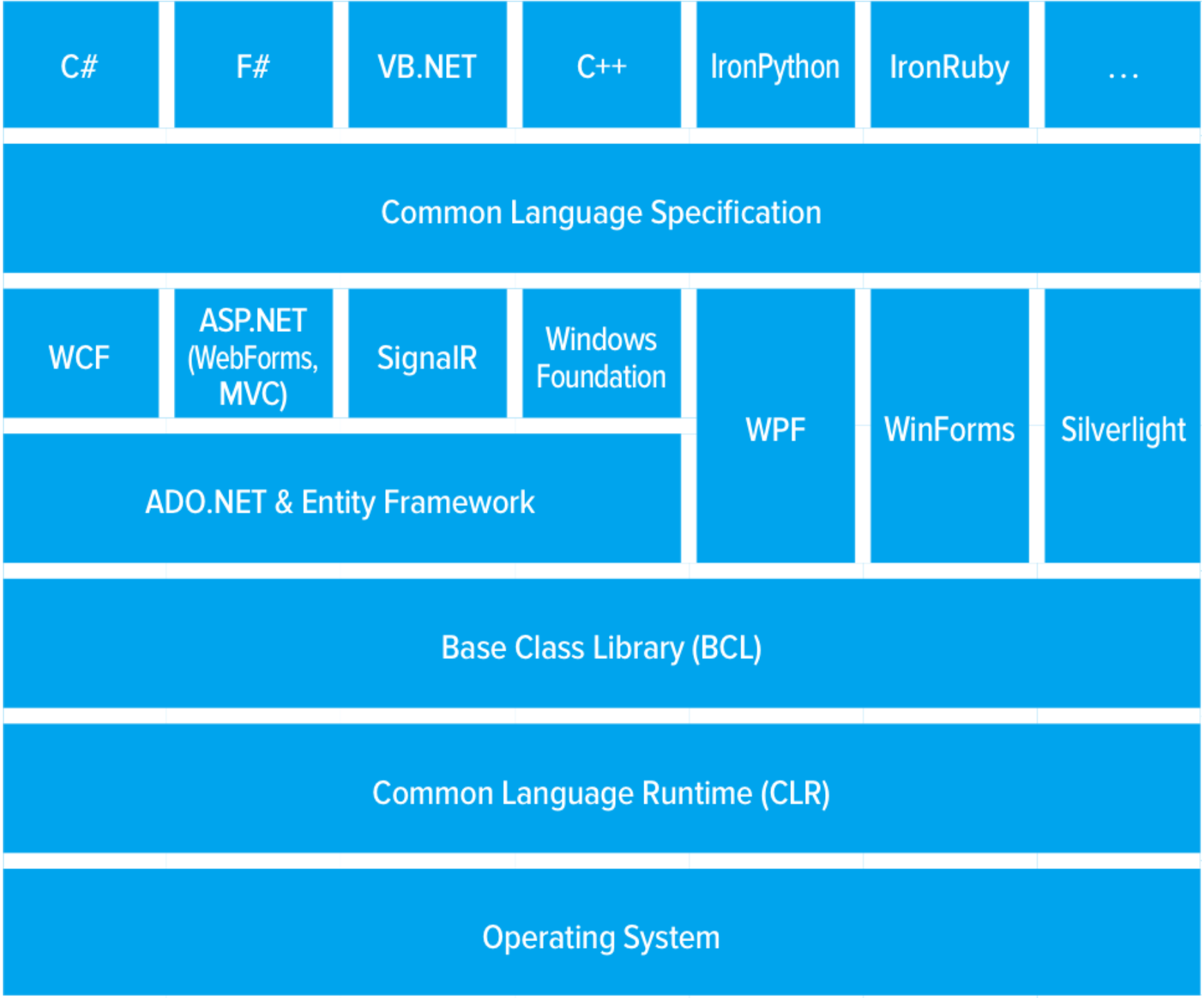
.NET (C#)

ASP.NET

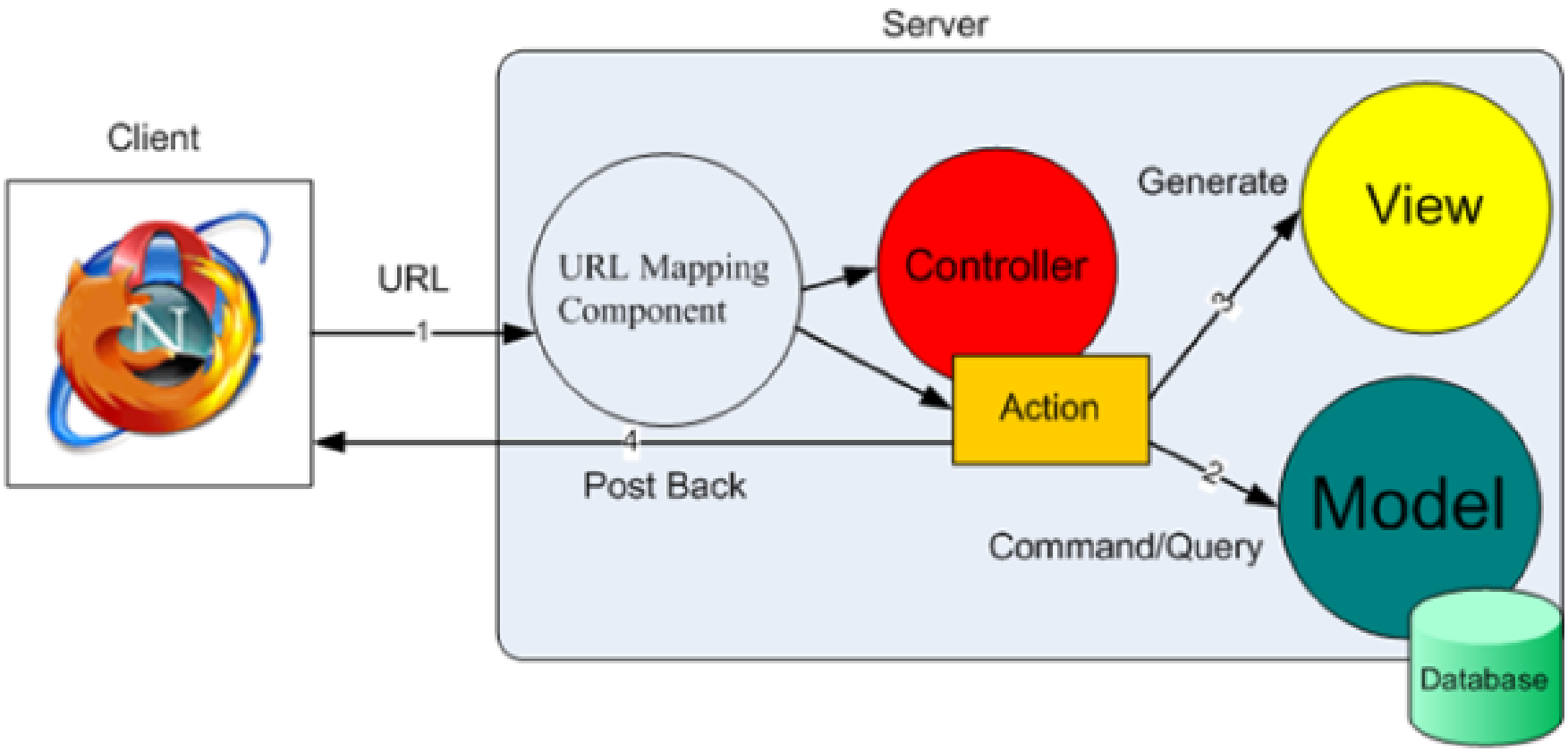
ENTITY FRAMEWORK


C#
F#
VB.NET
C++
IronPython
IronRuby
...
Common Language Specification
WCF
ASP.NET (WebForms, MVC)
SignalR
Windows Foundation
WPF
WinForms
Silverlight
ADO.NET & Entity Framework
Base Class Library (BCL)
Common Language Runtime (CLR)
Operating System

MVC

Model – View – Controller

MVC Architecture



Create Project MVC

New Project

- Recent
- Installed
  - Templates
    - Visual Basic
    - Visual C#
    - Visual C++
    - Visual F#
    - SQL Server
    - TypeScript
    - Python
    - Other Project Types
  - Samples
- Online

.NET Framework 4.5 | Sort by: Default | Search Installed Templates (Ctrl+E)

Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
ASP.NET Web Application	Visual C#
Class Library	Visual C#
Portable Class Library	Visual C#
Silverlight Application	Visual C#
Silverlight Class Library	Visual C#
WCF Service Application	Visual C#
LightSwitch Desktop Application	Visual C#
LightSwitch HTML Application	Visual C#

Click here to go online and find templates.

Type: Visual C#

A project template for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Name: MVC-BasicTutorials

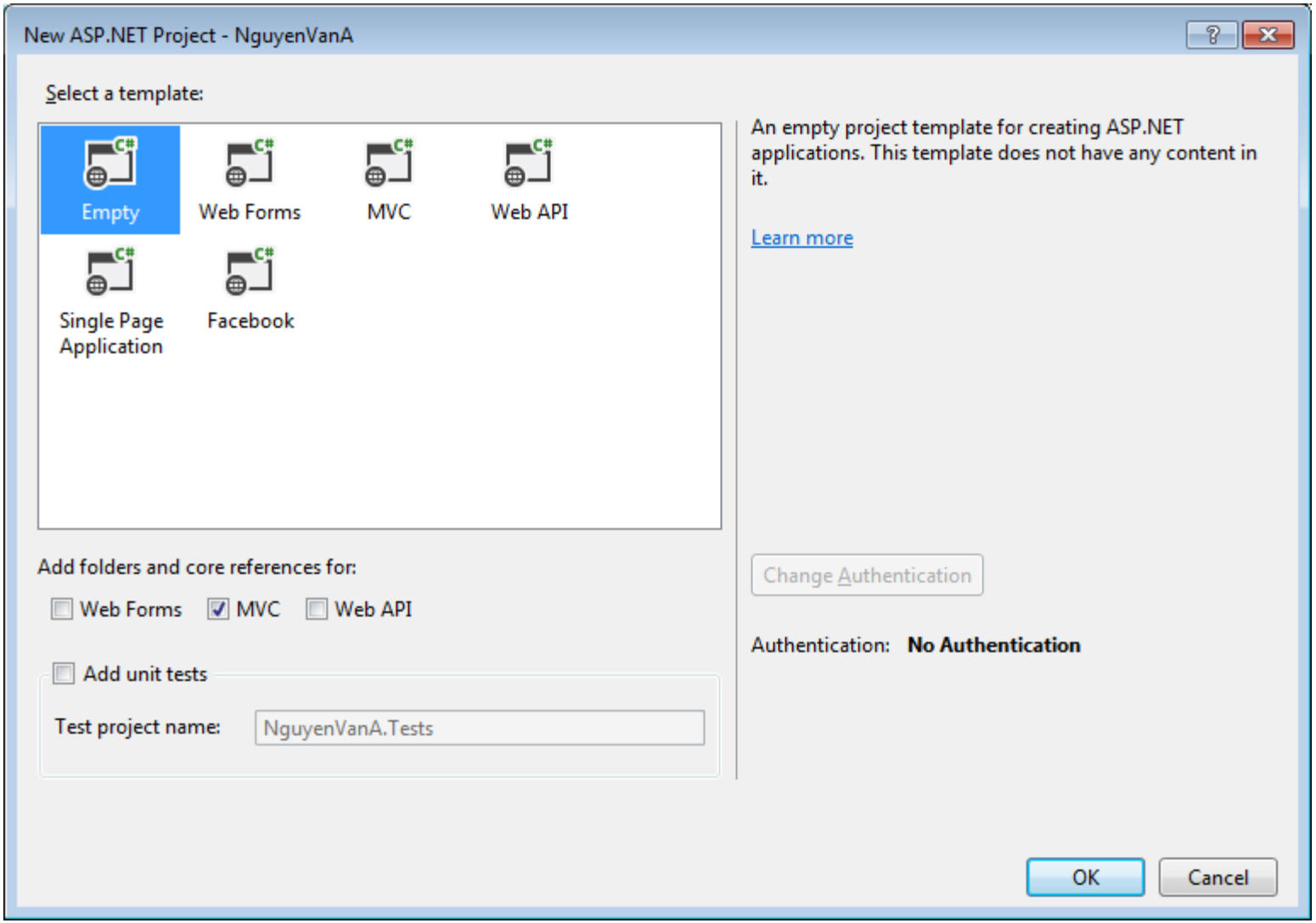
Location: D:\ Browse...

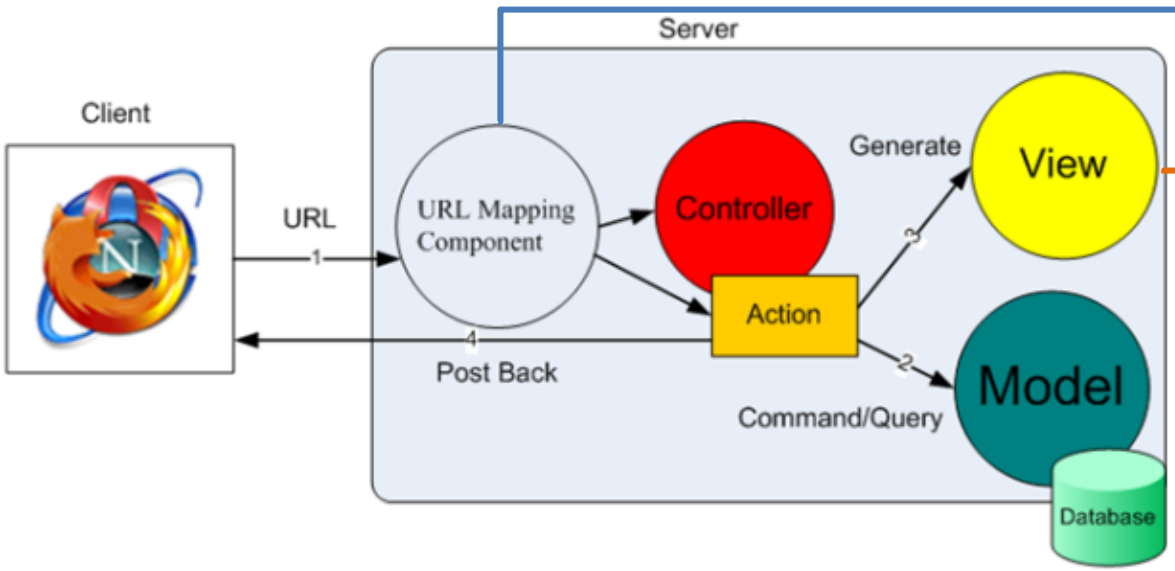
Solution name: MVC-BasicTutorials

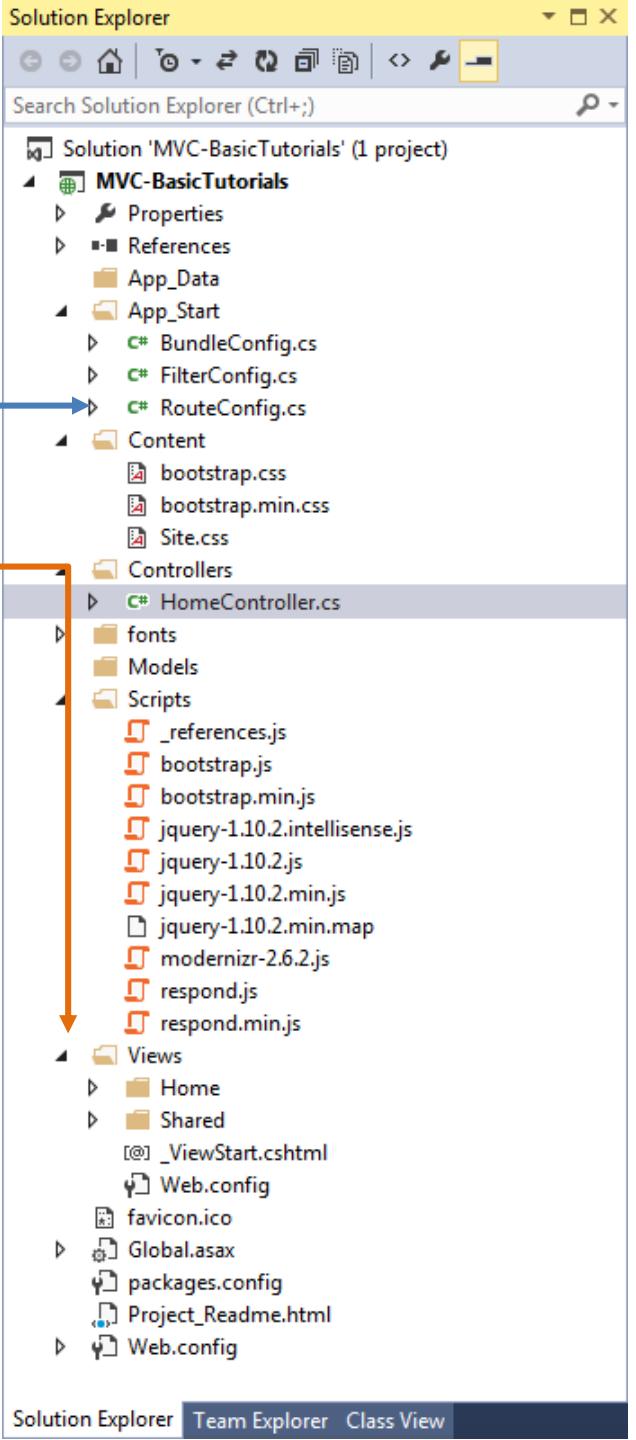
☑ Create directory for solution

☐ Add to source control

OK Cancel


New ASP.NET Project - NguyenVanA
Select a template:
Empty
Web Forms
MVC
Web API
Single Page Application
Facebook
An empty project template for creating ASP.NET applications. This template does not have any content in it.
Learn more
Add folders and core references for:
Web Forms
MVC
Web API
Change Authentication
Authentication: No Authentication
Add unit tests
Test project name:
NguyenVanA.Tests
OK
Cancel


Client
URL
1
Server
URL Mapping Component
Controller
Action
Generate
View
3
2
Model
Command/Query
Database
4
Post Back

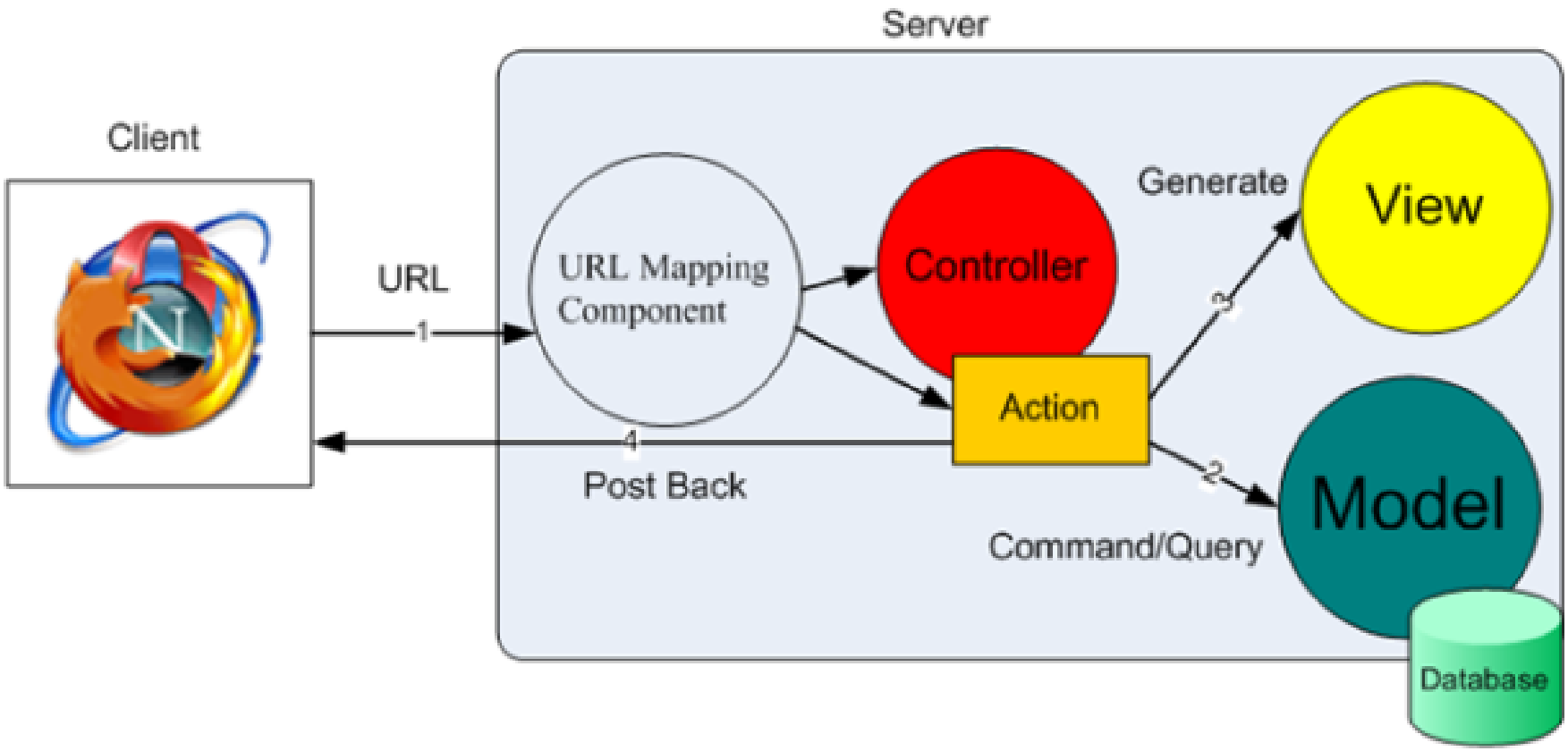

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MVC-BasicTutorials' (1 project)
MVC-BasicTutorials
Properties
References
App_Data
App_Start
BundleConfig.cs
FilterConfig.cs
RouteConfig.cs
Content
bootstrap.css
bootstrap.min.css
Site.css
Controllers
HomeController.cs
fonts
Models
Scripts
_references.js
bootstrap.js
bootstrap.min.js
jquery-1.10.2.intellisense.js
jquery-1.10.2.js
jquery-1.10.2.min.js
jquery-1.10.2.min.map
modernizr-2.6.2.js
respond.js
respond.min.js
Views
Home
Shared
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config
Solution Explorer
Team Explorer
Class View

ASP.NET (C#)

Controller

In this article, I'll explain an easy but an important concept of how to use – Controler in ASP.NET.

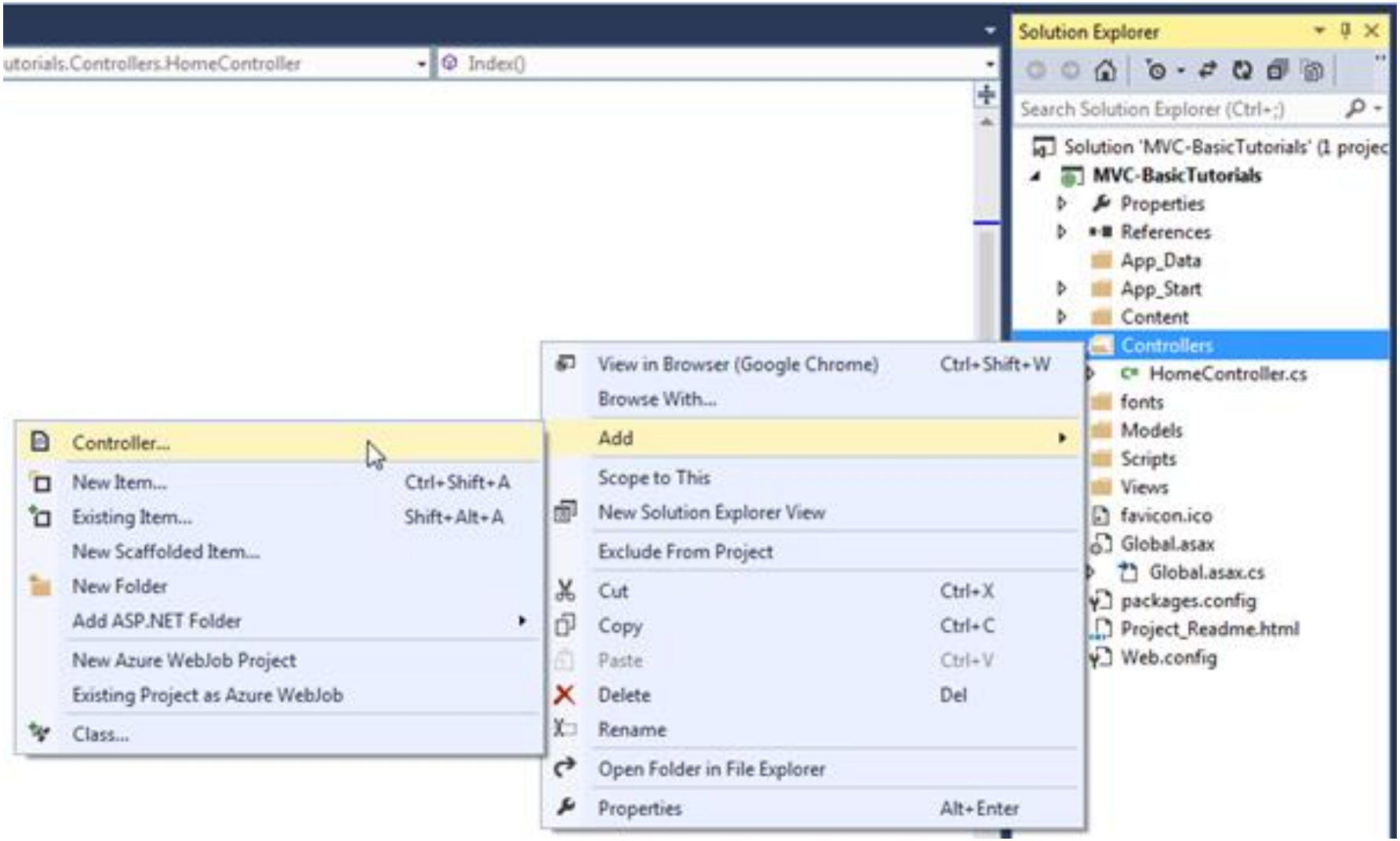
MVC Architecture



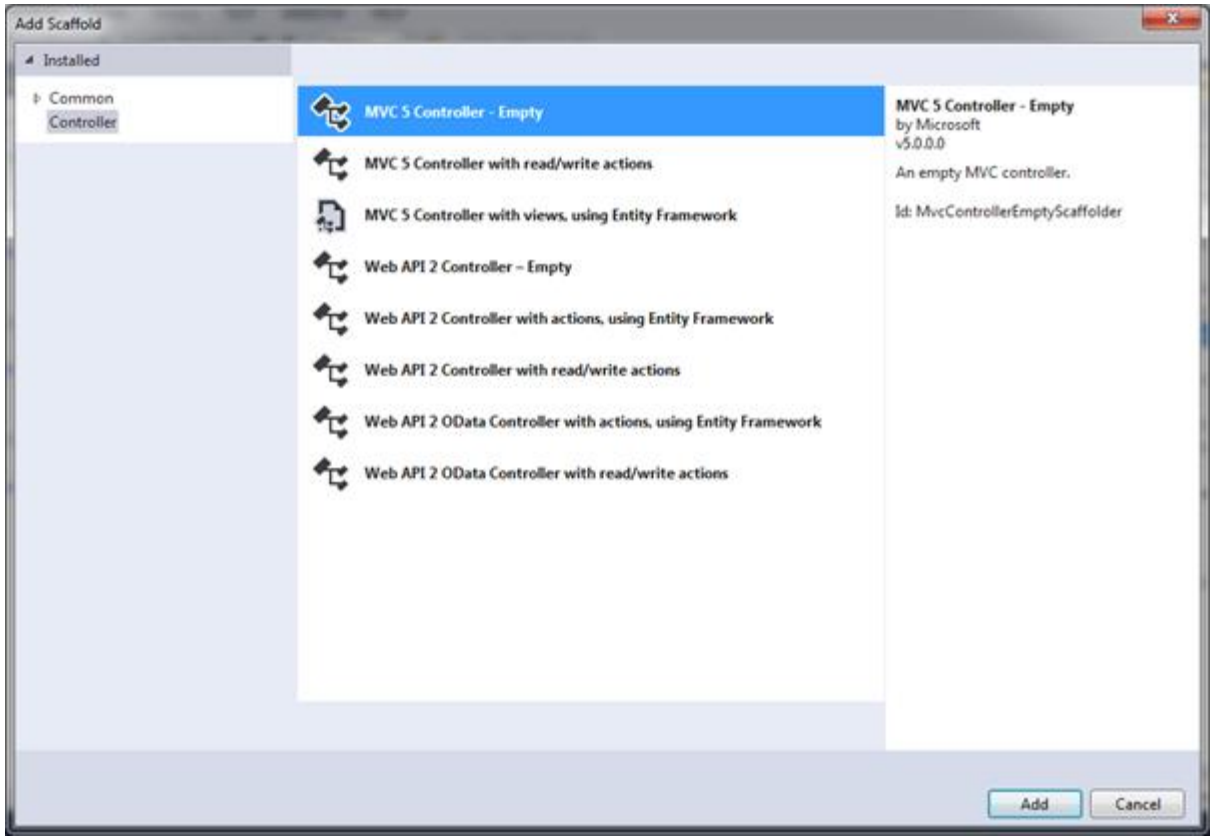
Controller

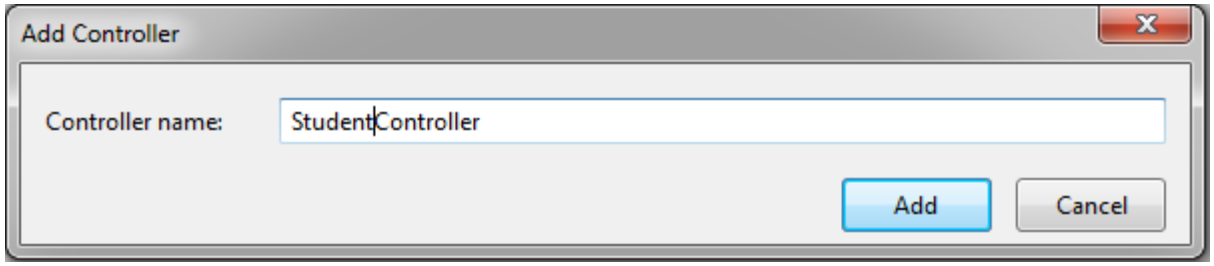
- The Controller in MVC architecture handles any incoming URL request.
- Controller is a class, derived from the base class System.Web.Mvc.Controller
- Controller class contains public methods called **Action** methods

Adding a New Controller



Adding a New Controller





Controller Example

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MVC_BasicTutorials.Controllers
{
  public class StudentController : Controller
  {
    // GET: Student
    public ActionResult Index()
    {
      return View();
    }
  }
}
```

ASP.NET (C#)

Action

In this article, I'll explain an easy but an important concept of how to use – Action in ASP.NET.

Action method

- All the public methods of a Controller class are called Action methods. They are like any other normal methods with the following restrictions:
  - Action method must be public. It cannot be private or protected
  - Action method cannot be overloaded
  - Action method cannot be a static method.

Example of Index action method of StudentController

Student Controller class

Base Controller class

```
public class StudentController : Controller
{
    // GET: Student
    public ActionResult Index()
    {
        return View();
    }
}
```

Return type

Action method

View() defined in base Controller class

Default Action Method

- Every controller can have default action method as per configured route in RouteConfig class
- By default, Index is a default action method for any controller

```
routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action =
"Index", id = UrlParameter.Optional }
            );
```

Action Method Parameters

```
[HttpPost]
public ActionResult Edit(Student std)
{
// update student to the database

      return RedirectToAction("Index");
}
[HttpDelete]
public ActionResult Delete(int id)
{
// delete student from the database whose id matches
with specified id
      return RedirectToAction("Index");
}
```

Invoked Action Method

http:// <domain_name> /<controller_name>/<actionmethod_name>

```
public ActionResult GetByNameList()
{
    return View();
}
```

*http://localhost/student/*GetByNameList

Invoked Action Method

http:// <domain_name> /<controller_name>/<actionmethod_name>

```
public ActionResult GetNameById(int id){
    return View();
}
```

*http://localhost/student/*getbynamelist/9

Invoked Action Method

http:// <domain_name> /<controller_name>/<actionmethod_name>/parameter

```
public ActionResult DetailsValType(int id, bool sortAscending)
{
    return GetDataAndResult(id, sortAscending);
}
```

*http://localhost/student/***detailsvaltype/1?sortAscending=false**

Action Selectors

- Action selector is the attribute that can be applied to the action methods. It helps routing engine to select the correct action method to handle a particular request.
  - ActionName
  - NonAction
  - ActionVerbs

Action Selectors Example

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    [ActionName("find")]
    public ActionResult GetById(int id)
    {
    // get student from the database
        return View();
    }
}
```

This action method will be invoked on *http://localhost/student/**find**/1*

ASP.NET (C#)

View

In this article, I'll explain an easy but an important concept of how to use – View in ASP.NET.

ASP.NET Razor - Markup

- Razor is not a programming language. It's a server side markup language.
- Razor is a markup syntax that lets you embed server-based code (C#) into web pages.

Razor Syntax

- Main Razor Syntax Rules for C#
- Razor code blocks are enclosed in @{ ... }
- Inline expressions (variables and functions) start with @
- Code statements end with semicolon
- Variables are declared with the var keyword
- Strings are enclosed with quotation marks
- C# code is case sensitive
- C# files have the extension .cshtml

Razor C# Example

```
<!-- Single statement block -->
@{ var myMessage = "Hello World"; }

<!-- Inline expression or variable -->
<p>The value of myMessage
is: @myMessage</p>

<!-- Multi-statement block -->
@{
var greeting = "Welcome to our site!";
var weekDay = DateTime.Now.DayOfWeek;
var greetingMessage = greeting + " Here in
Huston it is: " + weekDay;
}
<p>The greeting is: @greetingMessage</p>
```

ASP.NET Razor - C# Loops and Arrays

```
<html>
<body>
@for(var i = 10; i < 21; i++)
    {<p>Line @i</p>}
</body>
</html>
```

ASP.NET Razor - C# Loops and Arrays

```
<html>
<body>
@{
var i = 0;
while (i < 5)
    {
    i += 1;
    <p>Line @i</p>
    }
}
</body>
</html>
```

ASP.NET Razor - C# Loops and Arrays

```
@{
string[] members = {"Jani", "Hege", "Kai", "Jim"};
int i = Array.IndexOf(members, "Kai")+1;
int len = members.Length;
string x = members[2-1];
}
<html>
<body>
<h3>Members</h3>
@foreach (var person in members)
{
      <p>@person</p>
}
<p>The number of names in Members are @len</p>
<p>The person at position 2 is @x</p>
<p>Kai is now in position @i</p>
</body>
</html>
```

ASP.NET Razor - C# Logic Conditions

```
@{var price=20;}
<html>
<body>
@if (price>30)
{
  <p>The price is too high.</p>
}
else
{
  <p>The price is OK.</p>
}
</body>
</html>
```

ASP.NET Razor - C# Logic Conditions

```
@{
var weekday=DateTime.Now.DayOfWeek;
var day=weekday.ToString();
var message="";
}
<html>
<body>
@switch(day)
{
case "Monday":
    message="This is the first weekday.";
    break;
case "Thursday":
    message="Only one day before weekend.";
    break;
case "Friday":
    message="Tomorrow is weekend!";
    break;
default:
    message="Today is " + day;
    break;
}
<p>@message</p>
</body>
</html>
```

ASP.NET (C#)

ViewBag - ViewData

In this article, I'll explain an easy but an important concept of how to use – **ViewBag - ViewData** user in ASP.NET.

ViewBag

- ViewBag can be useful when you want to transfer temporary data (which is not included in model) from the controller to the view

```
public ActionResult Index()
{
  ViewBag.TotalStudents = 100;
  return View();
}
```

```
<label>TotalStudents:</label> @ViewBag.TotalStudents
```

ViewData

- ViewData is similar to ViewBag. It is useful in transferring data from Controller to View.
- ViewData is a dictionary which can contain key-value pairs where each key must be string

```
public ActionResult Index()
{
   ViewData["Name"] = "Nguyen Van A";
   ViewData["Age"] = 20;

   return View();
}
```

```
<label>Student Name:</label> @ViewData ["Name"]
<label>Student Age:</label> @ViewData ["Age"]
```

ASP.NET (C#)

HTML Helpers

In this article, I'll explain an easy but an important concept of how to use – **ViewBag - ViewData** user in ASP.NET.

```
@model IEnumerable<MVC_BasicTutorials.Models.Student>

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.StudentName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Age)
        </th>
        <th></th>
    </tr>
```

Extension method for HtmlHelper

HtmlHelper

ASP.NET (C#)

Article View

In this article, I'll explain an easy but an important concept of how to use – **View** user in ASP.NET.

Add New Project

- Recent
- Installed
  - Visual Basic
  - Visual C#
    - Windows
    - Web
    - Office/SharePoint
    - Cloud
    - LightSwitch
    - Reporting
    - Silverlight
    - Test
    - WCF
    - Workflow
  - Visual C++
  - Visual F#
  - SQL Server
  - TypeScript
  - Python
  - Other Project Types
- Online

.NET Framework 4.5 | Sort by: Default | Search Installed Templates (Ctrl+E)

Windows Forms Application	Visual C#
WPF Application	Visual C#
Console Application	Visual C#
ASP.NET Web Application	Visual C#
Class Library	Visual C#
Portable Class Library	Visual C#
Silverlight Application	Visual C#
Silverlight Class Library	Visual C#
WCF Service Application	Visual C#
LightSwitch Desktop Application	Visual C#
LightSwitch HTML Application	Visual C#
Get Windows Azure SDK for .NET	Visual C#

Click here to go online and find templates.

Type: Visual C#

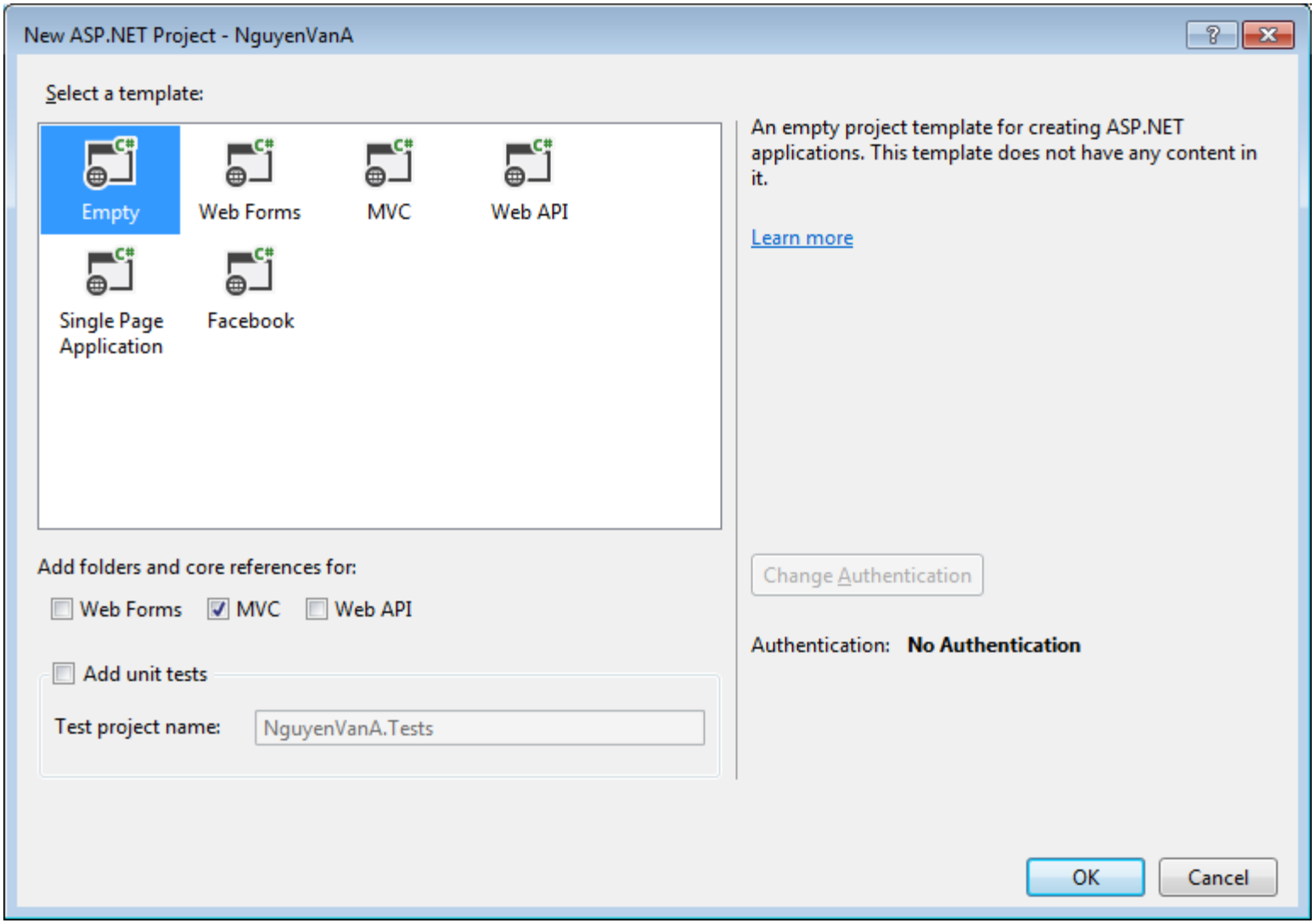
A project template for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Name: NguyenVanA

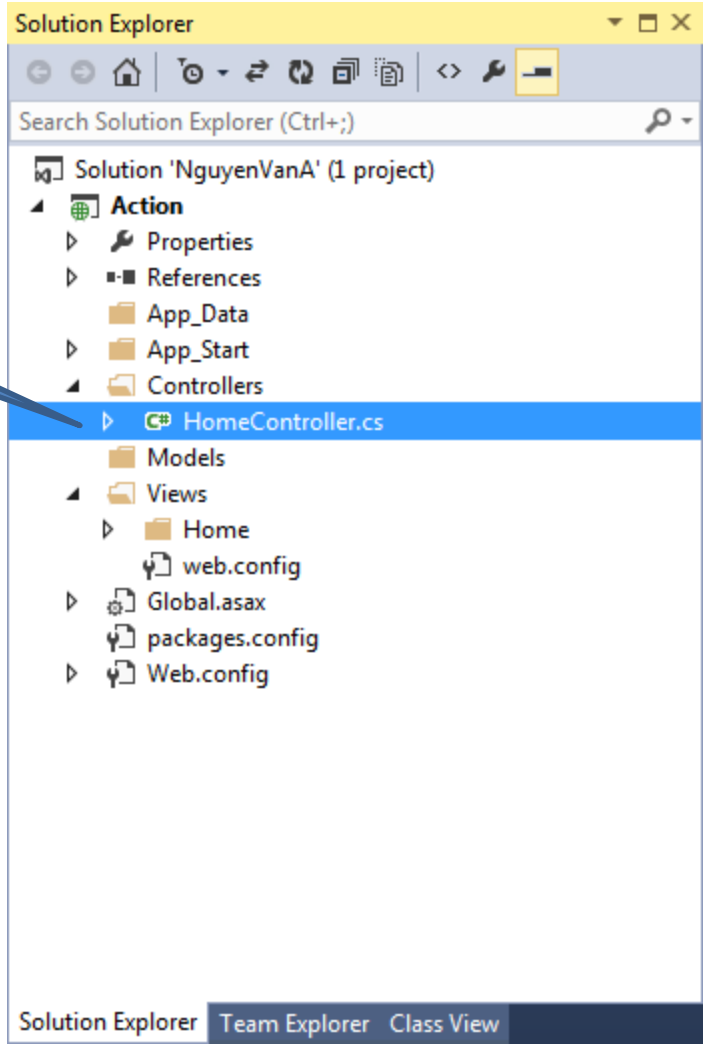
Location: D:\

Browse...

OK | Cancel


New ASP.NET Project - NguyenVanA
Select a template:
Empty
Web Forms
MVC
Web API
Single Page Application
Facebook
An empty project template for creating ASP.NET applications. This template does not have any content in it.
Learn more
Add folders and core references for:
Web Forms
MVC
Web API
Change Authentication
Authentication: No Authentication
Add unit tests
Test project name:
NguyenVanA.Tests
OK
Cancel

Edit class: HomeController



Model

```
public class HomeController : Controller
{
        public ActionResult Index(){
            return View();
        }
        public ActionResult ItemA(){
            return View();
        }
        public ActionResult ItemB(){
            return View();
        }
        public ActionResult ItemC(){
            return View();
        }
        public ActionResult ParameterA(string id){
            ViewData["m_id"] = id;
            return View();
        }
        public ActionResult ParameterB(string id, string pid){

            return View();
        }
}
```

NguyenVanA - Microsoft Visual Studio (Administrator)

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP

Google Chrome | Debug

HomeController.cs | Index.cshtml | Your ASP.NET application | web.config

WebApplication1.Controllers.HomeController | Index()

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace WebApplication1.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult Ite
        {
            return View();
        }
        public ActionResult Ite
        {
            return View();
        }
        public ActionResult Ite
        {
            return View();
        }

        public ActionResult Par
        {

            return View();
        }

        public ActionResult Par
        {
```

- Go To View — Ctrl+M, Ctrl+G
- Add View...
- Refactor
- Organize Usings
- Insert Snippet... — Ctrl+K, Ctrl+X
- Surround With... — Ctrl+K, Ctrl+S
- Peek Definition — Alt+F12
- Go To Definition — F12
- Find All References — Shift+F12
- View Call Hierarchy — Ctrl+K, Ctrl+T
- Breakpoint
- Run To Cursor — Ctrl+F10
- Run Flagged Threads To Cursor
- Cut — Ctrl+X
- Copy — Ctrl+C
- Paste — Ctrl+V
- Outlining

Solution Explorer

Search Solution Explorer (Ctrl+;)

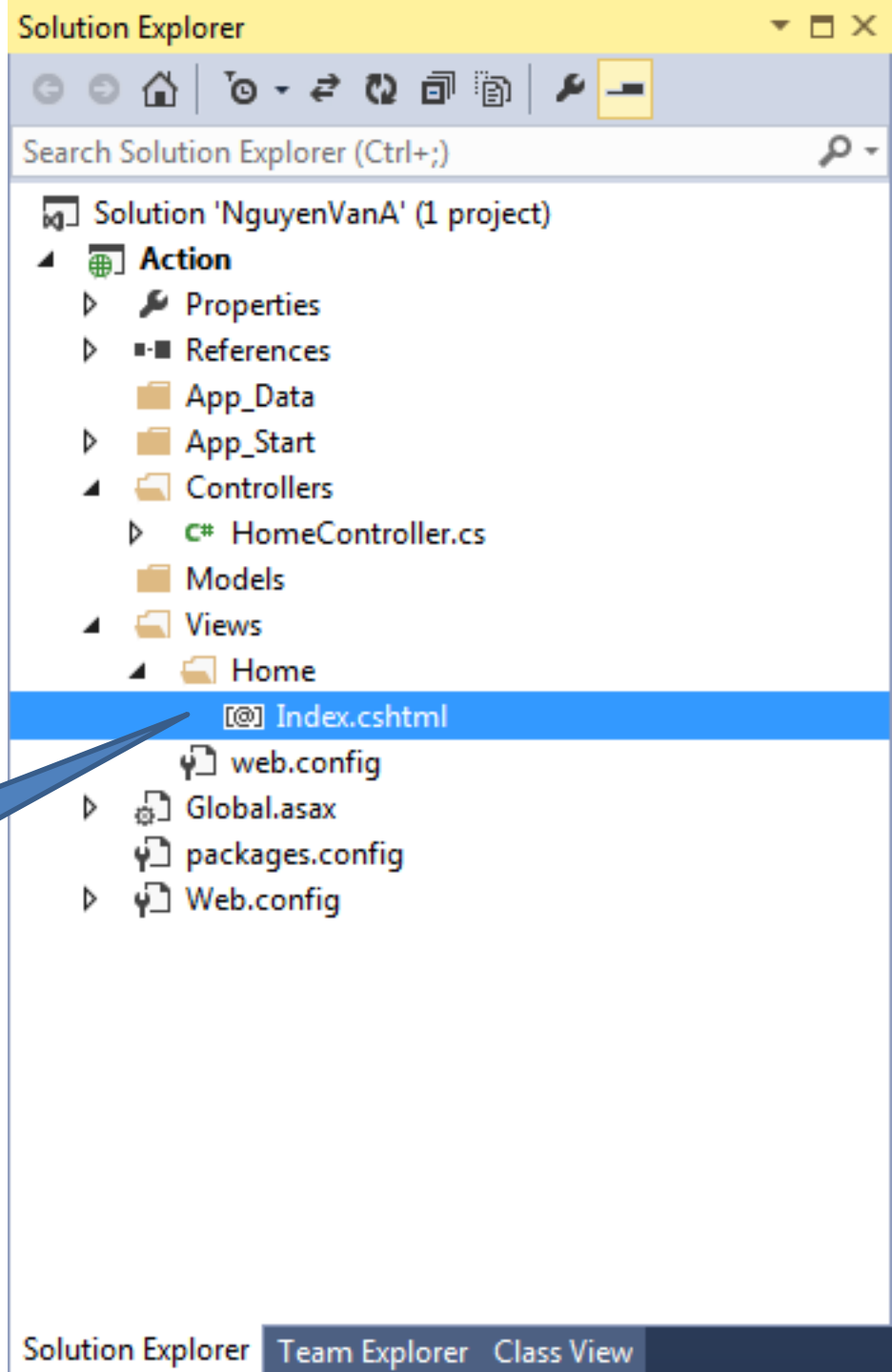
- Solution 'NguyenVanA' (1 project)
  - Action
    - Properties
    - References
    - App_Data
    - App_Start
    - Controllers
      - HomeController.cs
    - Models
    - Views
      - Home
      - web.config
    - Global.asax
    - packages.config
    - Web.config

Solution Explorer | Team Explorer | Class View

Output

Ready | Ln 11 | Col 32 | Ch 32 | INS

Clear all and Edit file: Index.html



```
@{
    ViewBag.Title = "Home Page";
}

<ul>
    <li>@Html.ActionLink("Item 1A", "ItemA", "Home")</li>
    <li>@Html.ActionLink("Item 1B", "ItemB", "Home")</li>
</ul>

<ul>
    <li>@Html.ActionLink("Parameter 1A", "ParameterA", new { @id =
"hello" },null)</li>
    <li>@Html.ActionLink("Parameter 1B", "ParameterB", new { @id =
"hello", @pid = "love" }, null)</li>
</ul>
```

localhost:50001/Home/Index

- Item 1A
- Item 1B

- Parameter 1A
- Parameter 1B

F5...Run...

The view 'ItemA' or its m

localhost:50001/Home/ItemA

Server Error in '/' Application.

The view 'ItemA' or its master was not found or no view engine supports the searched locations. The following locations were searched:
~/Views/Home/ItemA.aspx
~/Views/Home/ItemA.ascx
~/Views/Shared/ItemA.aspx
~/Views/Shared/ItemA.ascx
~/Views/Home/ItemA.cshtml
~/Views/Home/ItemA.vbhtml
~/Views/Shared/ItemA.cshtml
~/Views/Shared/ItemA.vbhtml

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.InvalidOperationException: The view 'ItemA' or its master was not found or no view engine supports the searched locations. The following locations were searched:
~/Views/Home/ItemA.aspx
~/Views/Home/ItemA.ascx
~/Views/Shared/ItemA.aspx
~/Views/Shared/ItemA.ascx
~/Views/Home/ItemA.cshtml
~/Views/Home/ItemA.vbhtml
~/Views/Shared/ItemA.cshtml
~/Views/Shared/ItemA.vbhtml

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'NguyenVanA' (1 project)
  - Action
    - Properties
    - References
    - App_Data
    - App_Start
    - Controllers
      - HomeController.cs
    - Models
    - Scripts
    - Views
      - Home
        - Index.cshtml
        - ItemA.cshtml
        - ItemB.cshtml
        - ItemC.cshtml
        - ParameterA.cshtml
        - ParameterB.cshtml
      - Shared
        - _Layout.cshtml
      - _ViewStart.cshtml
      - web.config
    - Global.asax
    - packages.config
    - Web.config

Solution Explorer | Team Explorer | Class View

F5...Run...

ItemA - My ASP.NET App

localhost:50001/Home/ItemA

Application name

ItemA

© 2018 - My ASP.NET Application

ParameterA.cshtml

```
@{
    ViewBag.Title = "ParameterA";
}

@{
    var prod = (string)ViewData["m_id"];
}

<h2>ParameterA</h2>
<h1>ID: @prod</h1>
```

F5…Run…

Home Page - My ASP.NE

localhost:50001

Application name

- Item 1A
- Item 1B

- Parameter 1A
- Parameter 1B

© 2018 - My ASP.NET Application

ItemA - My ASP.NET App

localhost:50001/Home/ItemA

Application name

ItemA

© 2018 - My ASP.NET Application

ParameterA - My ASP.NE

localhost:50001/Home/ParameterA/hello

Application name

ParameterA

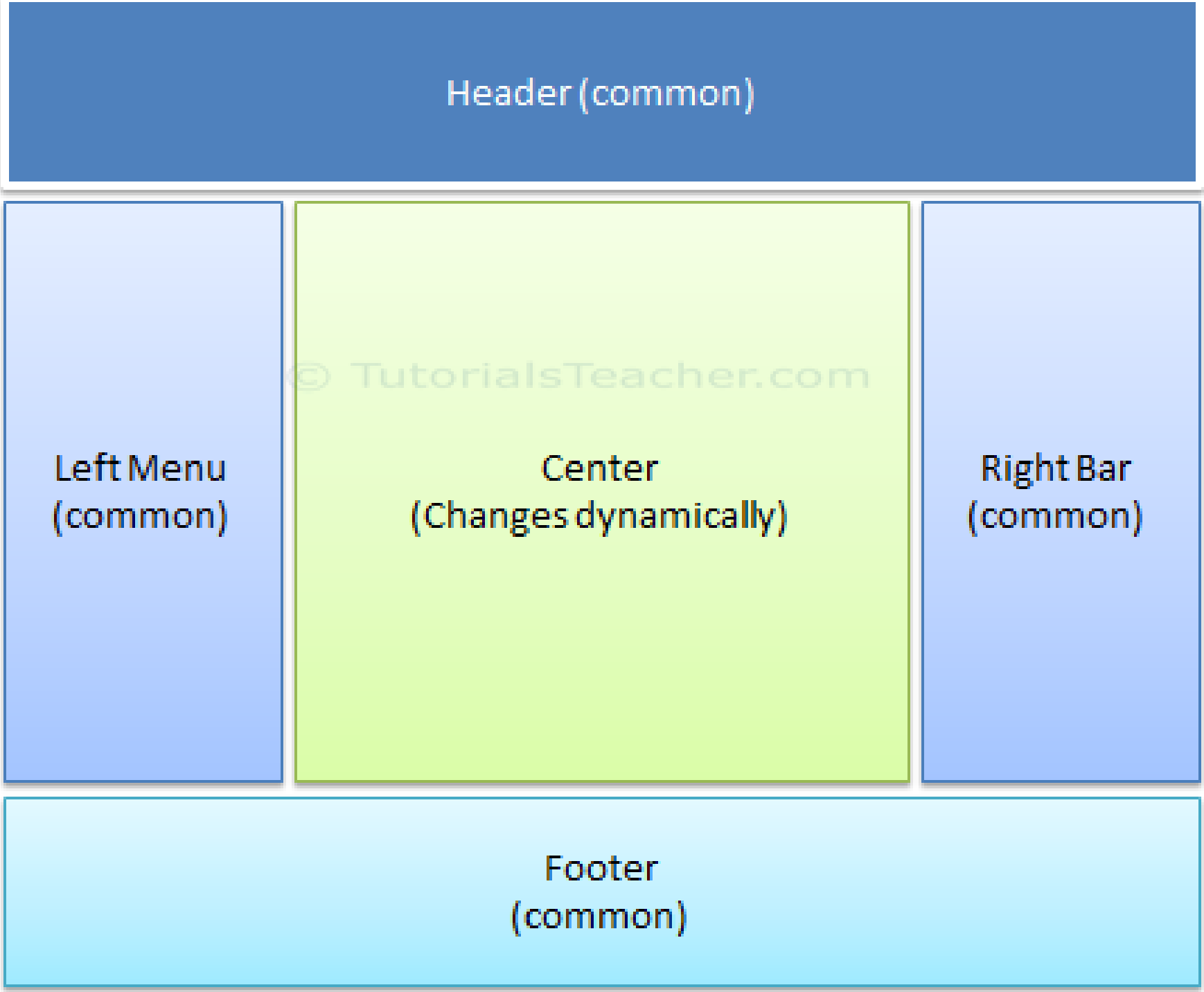
ID: hello

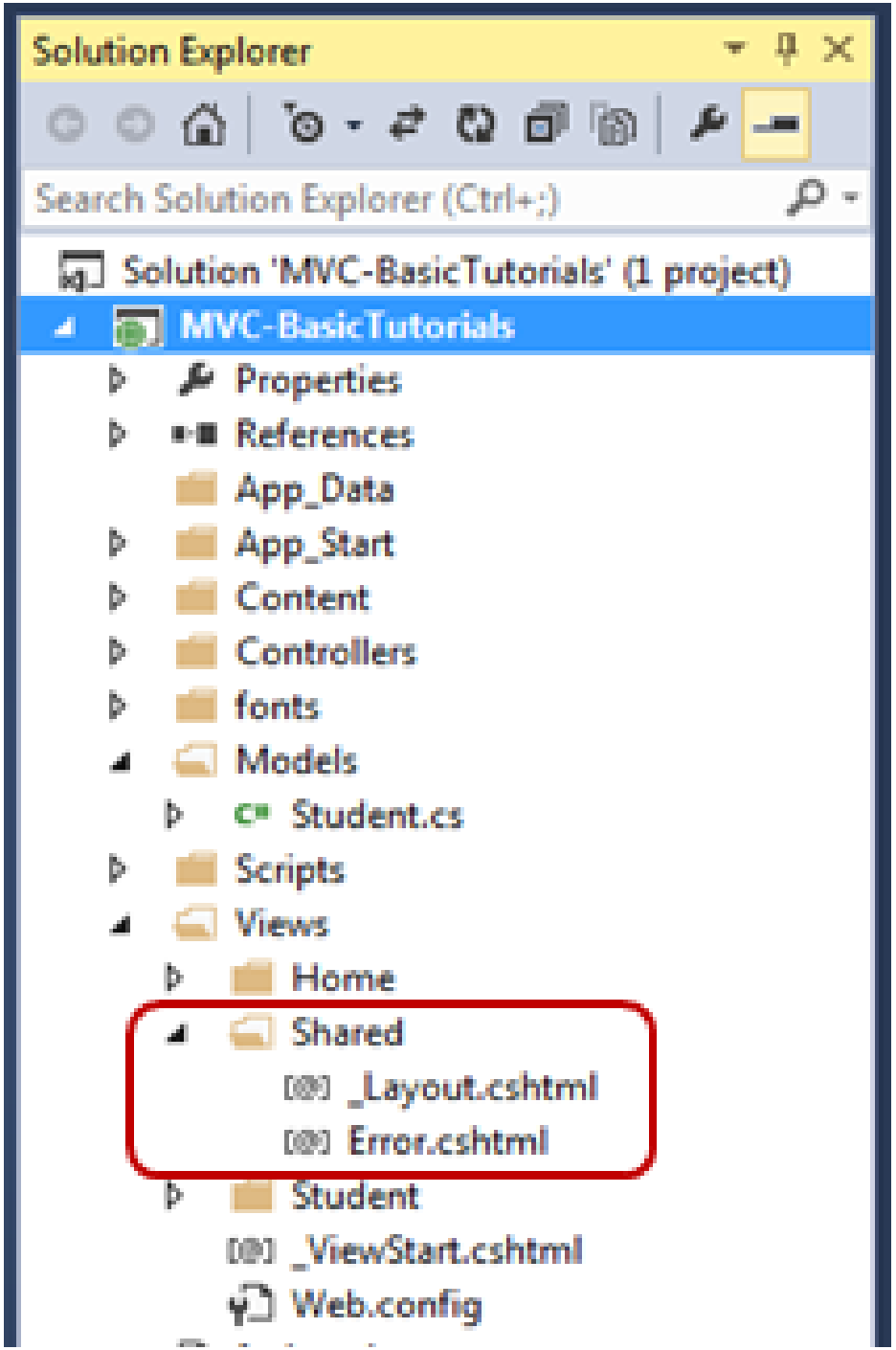
© 2018 - My ASP.NET Application

ASP.NET (C#)

Layout Article 01

In this article, I'll explain an easy but an important concept of how to use **Layout** in ASP.NET.




Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MVC-BasicTutorials' (1 project)
MVC-BasicTutorials
Properties
References
App_Data
App_Start
Content
Controllers
fonts
Models
Student.cs
Scripts
Views
Home
Shared
_Layout.cshtml
Error.cshtml
Student
_ViewStart.cshtml
Web.config

_ViewStart.cshtml

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```


Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'MVC-BasicTutorials' (1 project)
  - MVC-BasicTutorials
    - Properties
    - References
    - App_Data
    - App_Start
    - Content
    - Controllers
    - fonts
    - Models
    - Scripts
    - Views
      - Home
      - Shared
      - Student
      - _ViewStart.cshtml
      - Web.config
    - favicon.ico
    - Global.asax
    - packages.config
    - Project_Readme.html
    - Web.config

_ViewStart.cshtml

```
@{
    Layout = "~/Views/Shared/_myLayoutPage.cshtml";
}
```


Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'MVC-BasicTutorials' (1 project)
  - MVC-BasicTutorial
    - Properties
    - References
    - App_Data
    - App_Start
    - Content
    - Controllers
    - fonts
    - Models
    - Scripts
    - Views
      - Home
        - _ViewStart.cshtml
        - About.cshtml
        - Contact.cshtml
        - Index.cshtml
      - Shared
        - _Layout.cshtml
        - _myLayoutPage.cshtml
        - Error.cshtml
      - Student
      - _ViewStart.cshtml
      - Web.config

_myLayoutPage.cshtml

```
header {
height: 100px;
width: 100%;
background-color: red;
}

nav {
float: right;
width: 200px;
height: 250px;
background-color: darkgoldenrod;
}

.content {
background-color: aliceblue;
padding: 20px;
}

footer {
background-color: green;
width: 100%;
height: 50px;
float: right;
text-align: center;
}
```

_myLayoutPage.cshtml

```
<body style="background-color:burlywood">
    <header>
        <h1 style="text-align:center; color:bisque">BANXE.VN</h1>
    </header>

    <nav>
        <h3>Navigation</h3>
        @Html.ActionLink("Trang chủ", "About");<br />
        @Html.ActionLink("Ban xe", "Contact");<br />
        @Html.ActionLink("Tin tức", "Enquiry");<br />
        @Html.ActionLink("Tư vấn", "Home");<br />
        @Html.ActionLink("Đánh giá", "Purchase");<br />
        @Html.ActionLink("So sánh", "Purchase");<br />
    </nav>

    <div class="content">
        @RenderBody()
    </div>

    <footer>
        <h4>I am Footer.</h4>
    </footer>
</body>
```

Index.cshtml

```
@{
    ViewBag.Title = "Home Page";
    Layout = "~/Views/Shared/_myLayoutPage.cshtml";
}
```

F5…Run…

ASP.NET (C#)

Layout Article 02

In this article, I'll explain an easy but an important concept of how to use **Layout** in ASP.NET.

In the *Shared* folder, create a file named *_Header.cshtml*.

```
<div class="header">
    This is header text.
</div>
```

In the *Shared* folder, create a file named *_Footer.cshtml*.

```
<div class="footer">
    This is footer text.
</div>
```

Index.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <title>Main Page</title>
</head>
<body>
    <p>This is the content of the main page.</p>
</body>
</html>
```

In the *Shared* folder, create a file named *_Layout.cshtml*.

```
<!DOCTYPE html>
<html>
<head>
    <title>Structured Content </title>
</head>
<body>
        @RenderPage("~/Views/Shared/_Header.cshtml")
        @RenderBody()
        @RenderPage("~/Views/Shared/_Footer.cshtml")
</body>
</html>
```

In the ***View*** folder, create a file named _*ViewStart.cshtml*.

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

F5…Run…

ASP.NET (C#)

Model

In this article, I'll explain an easy but an important concept of how to use – Model in ASP.NET.

Model class Example

```
namespace MVC_BasicTutorials.Models
{
    public class Student
    {
        public int StudentId { get; set; }
        public string StudentName { get; set; }
        public int Age { get; set; }
    }
}
```

MVC

Integrate Model – View – Controller

ASP.NET (C#)

Article 01

In this article, I'll explain an easy but an important concept of how to use – **Model - View - Controller** user in ASP.NET.

New Project

Recent

Installed

Templates
- Visual Basic
- Visual C#
  - Windows Store
  - Windows
  - Web
  - Office/SharePoint
  - Cloud
  - LightSwitch
  - Reporting
  - Silverlight
  - Test
  - WCF
  - Workflow
- Visual C++
- Visual F#
- SQL Server
- TypeScript
- JavaScript
- Python
- Other Project Types

Online

.NET Framework 4.5.1 | Sort by: Default | Search Installed Templates (Ctrl+E)

ASP.NET Web Application — Visual C#

Click here to go online and find templates.

Type: Visual C#

A project template for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Name: HITC

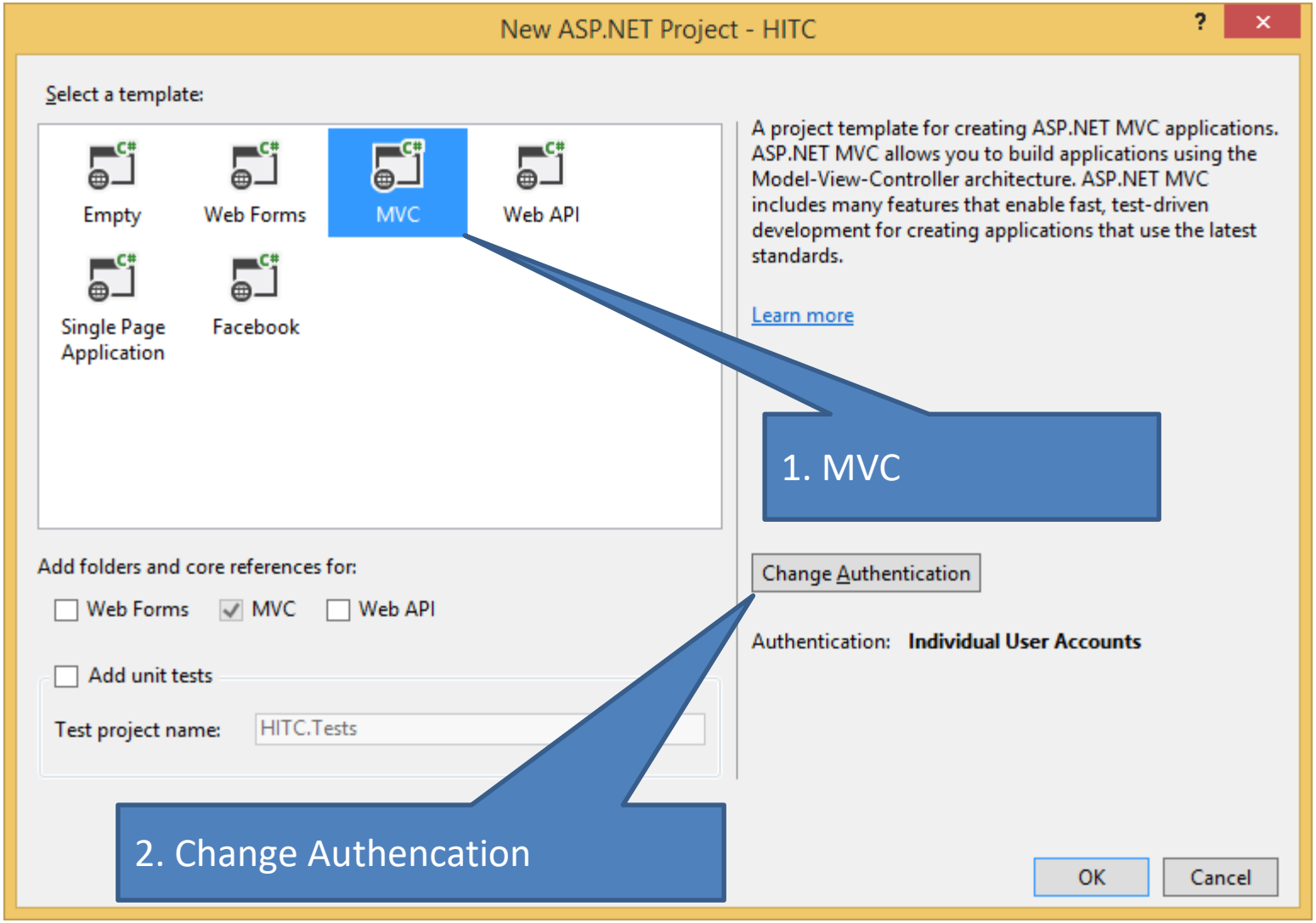
Location: F:\ Browse...

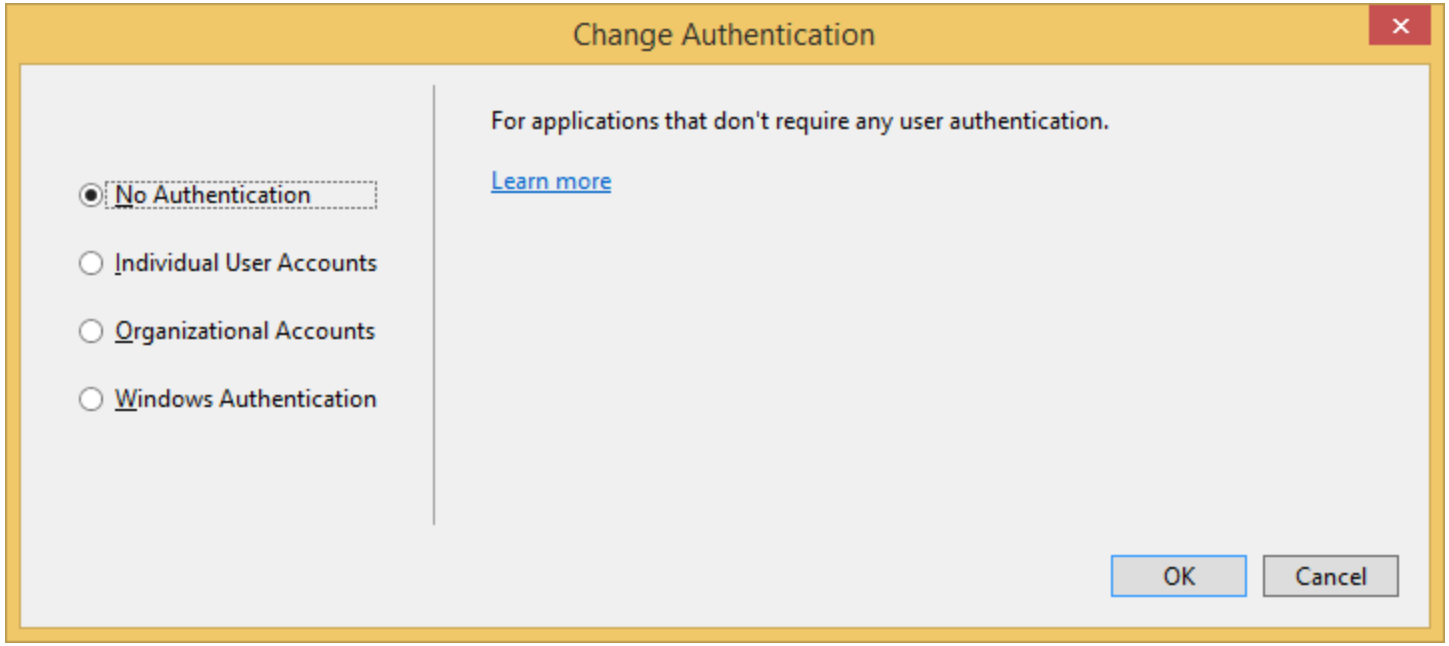
Solution name: HITC

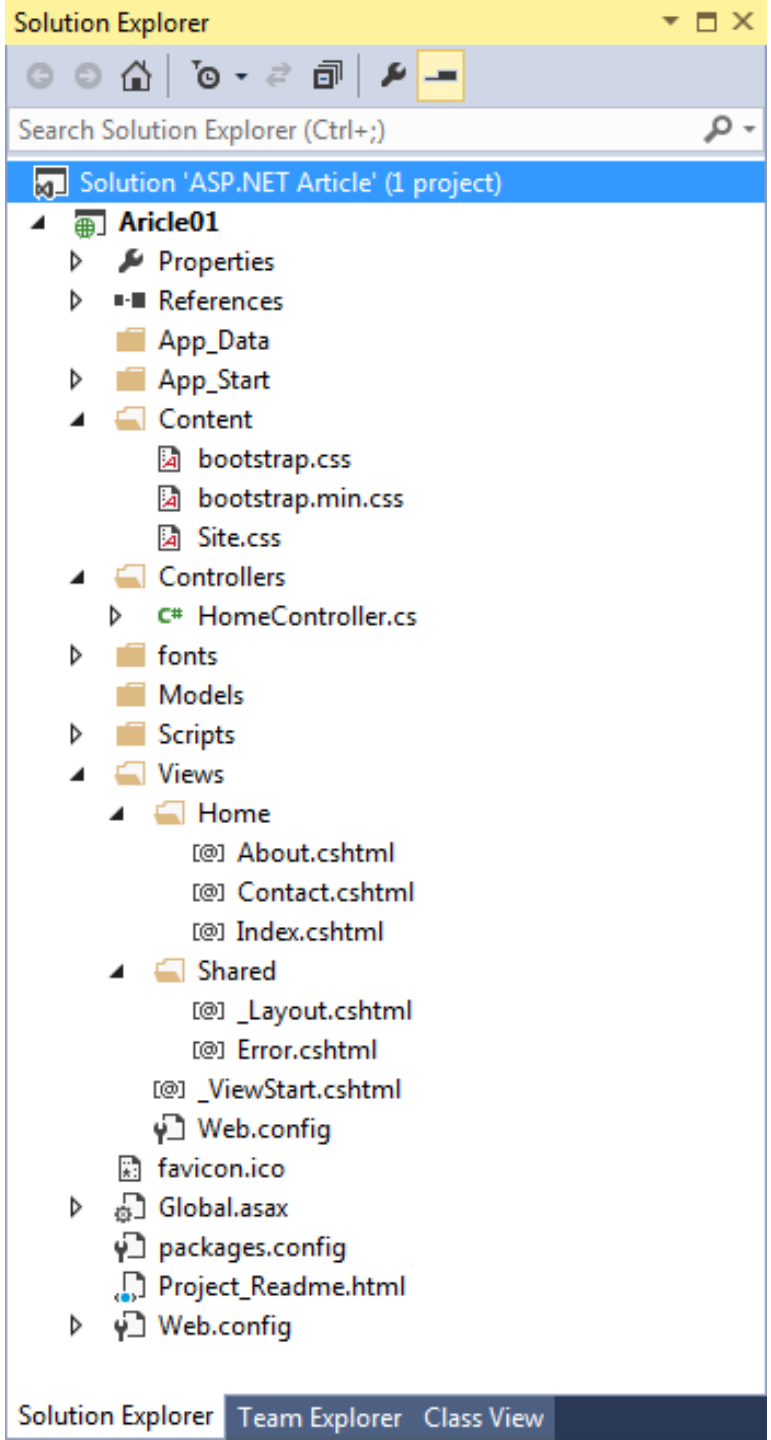
☑ Create directory for solution

☐ Add to source control

OK Cancel


New ASP.NET Project - HITC
Select a template:
Empty
Web Forms
MVC
Web API
Single Page Application
Facebook
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.
Learn more
1. MVC
Add folders and core references for:
Web Forms
MVC
Web API
Change Authentication
Authentication: Individual User Accounts
Add unit tests
Test project name:
HITC.Tests
2. Change Authencation
OK
Cancel

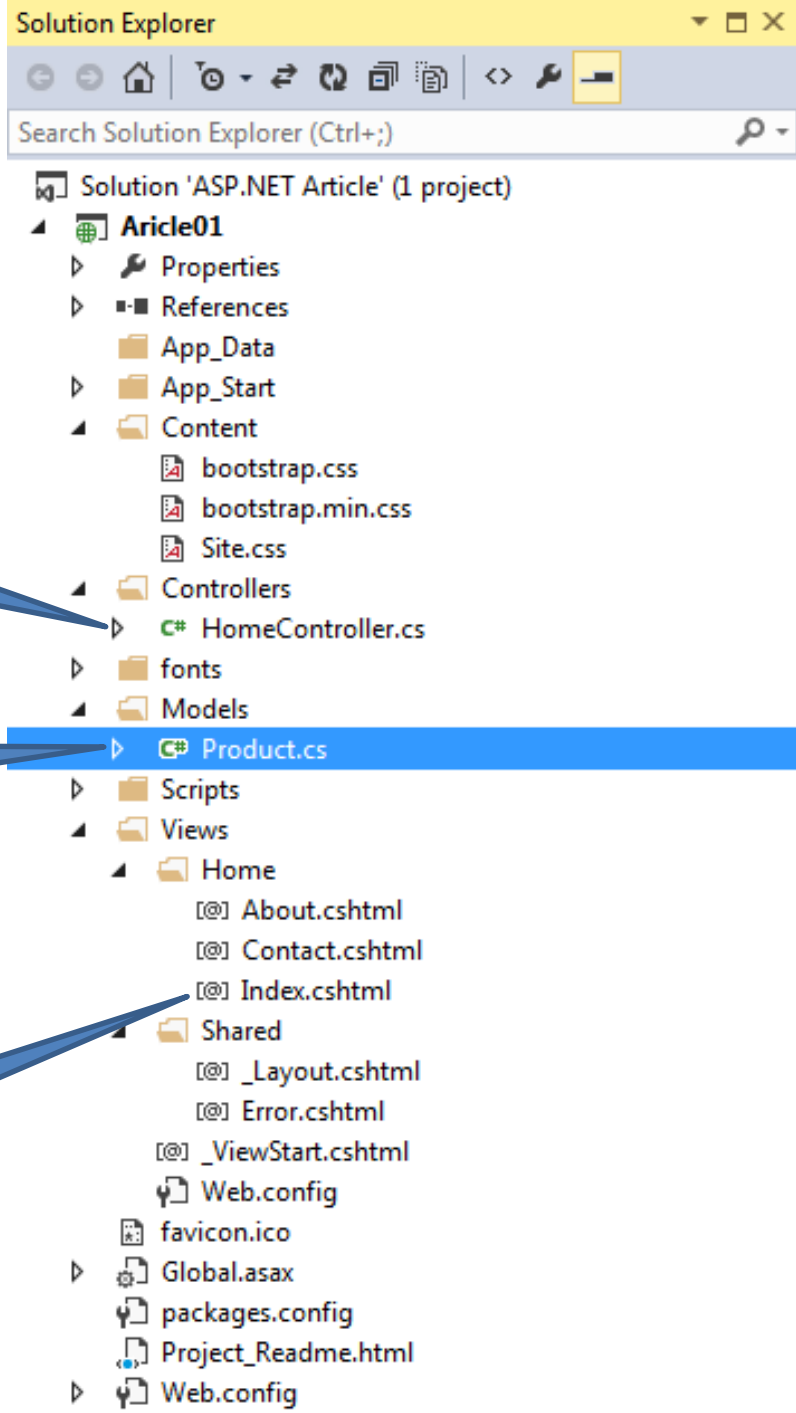

Change Authentication
No Authentication
Individual User Accounts
Organizational Accounts
Windows Authentication
For applications that don't require any user authentication.
Learn more
OK
Cancel


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ASP.NET Article' (1 project)
Aricle01
Properties
References
App_Data
App_Start
Content
bootstrap.css
bootstrap.min.css
Site.css
Controllers
HomeController.cs
fonts
Models
Scripts
Views
Home
About.cshtml
Contact.cshtml
Index.cshtml
Shared
_Layout.cshtml
Error.cshtml
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config
Solution Explorer
Team Explorer
Class View

Edit class: HomeController

Add new class: Product

Clear all and Edit file: Index.html



Model

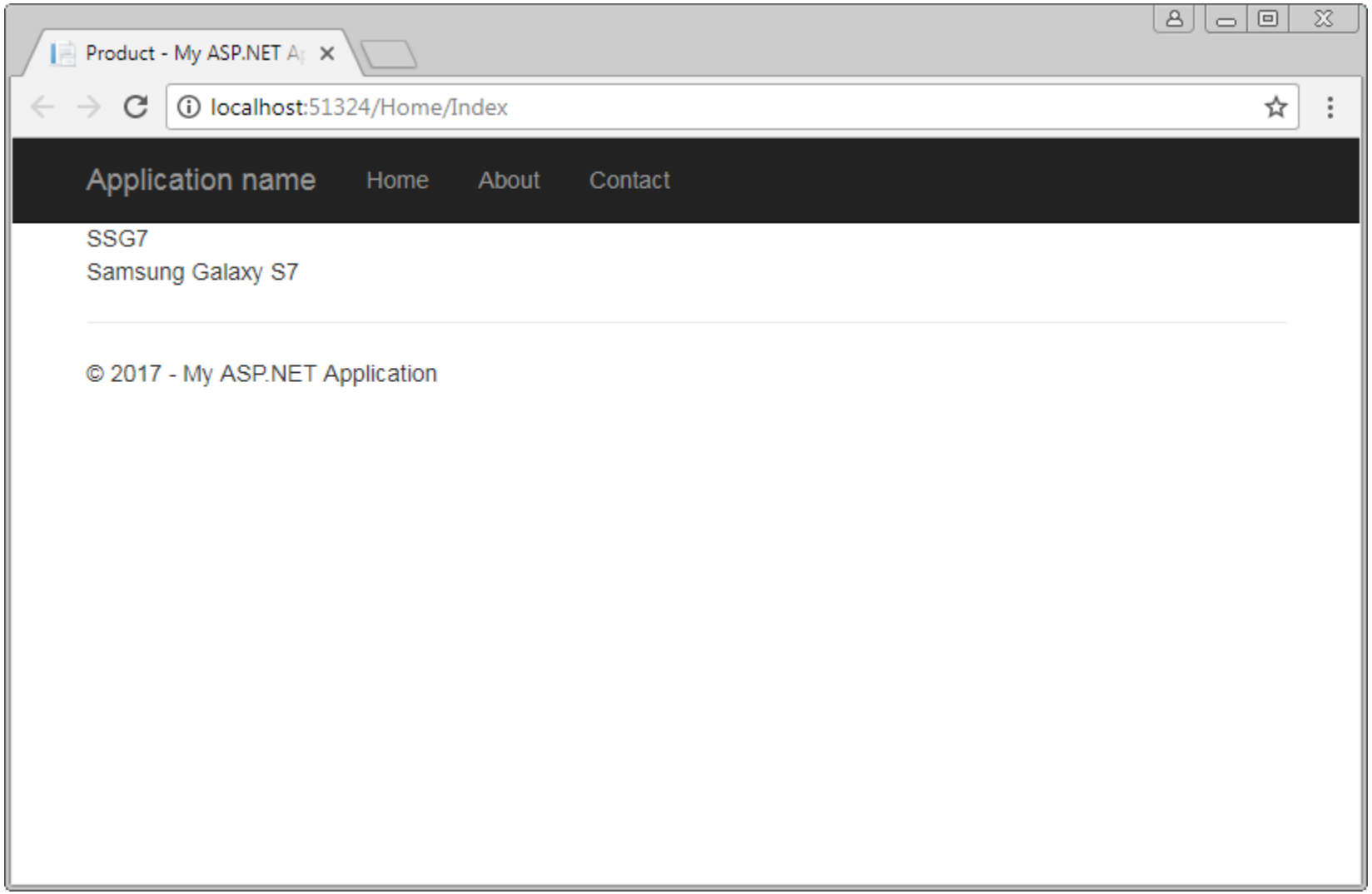
```
public class Product

{

        public string ID { get; set; }

        public string Name { get; set; }

}
```

```
public class HomeController : Controller
{
        public ActionResult Index()
        {
            Product pro = new Product();
            pro.ID = "SSG7";
            pro.Name = "Samsung Galaxy S7";
            return View(pro);
        }
}
```

```
@model Article01.Models.Product
@{
    ViewBag.Title = "Product";
}


@Html.DisplayFor(model => model.ID)
<br />
@Html.DisplayFor(model => model.Name)
```

Index.cshtml

Run… Test


Product - My ASP.NET A
localhost:51324/Home/Index
Application name
Home
About
Contact
SSG7
Samsung Galaxy S7
© 2017 - My ASP.NET Application

ASP.NET (C#)

Article 02

In this article, I'll explain an easy but an important concept of how to use **Model - View - Controller** user in ASP.NET.

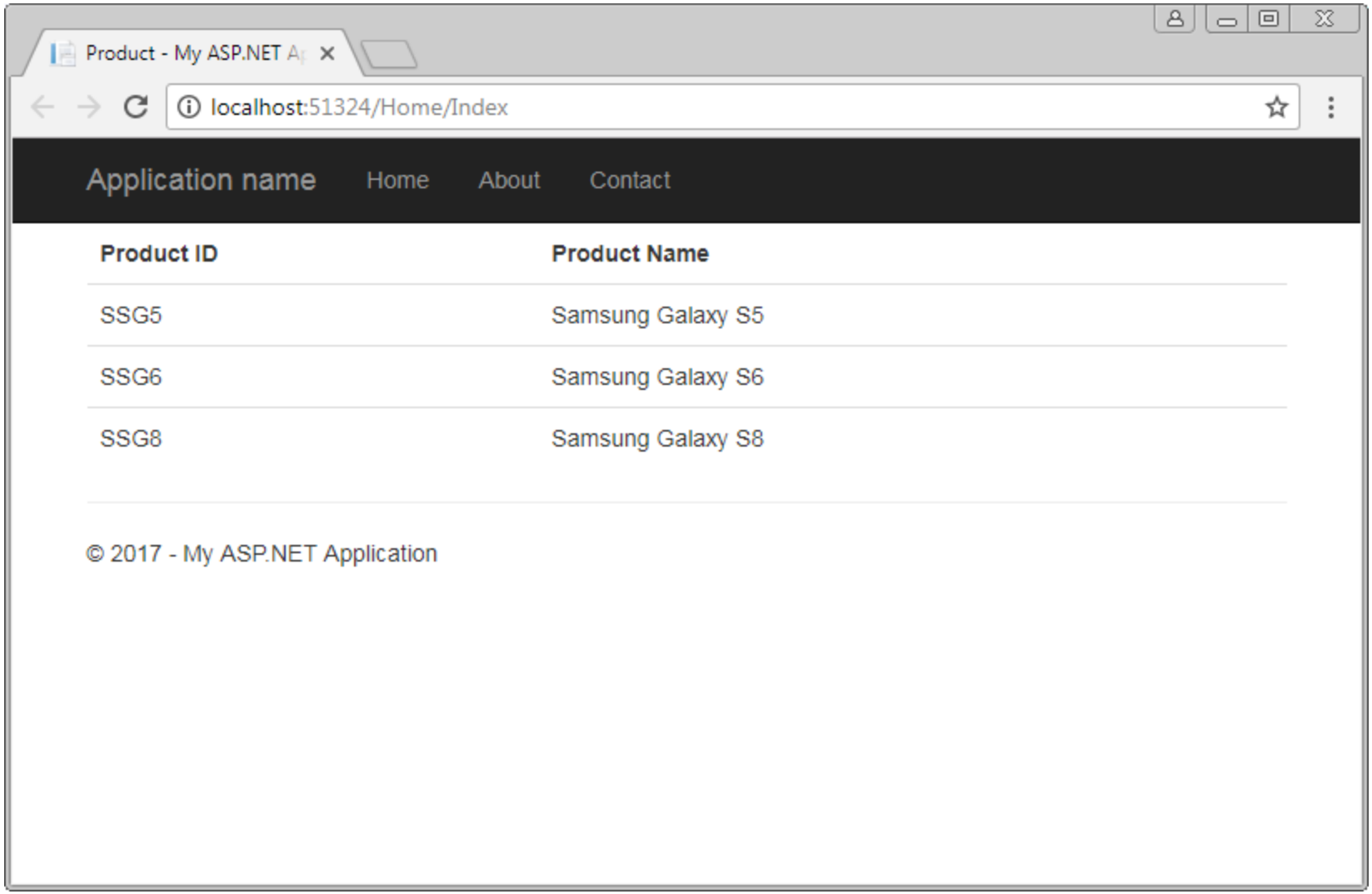
```
public ActionResult Index()
{
        List<Product> lstProduct = new List<Product>();

        Product pro;
        pro = new Product();
        pro.ID = "SSG5";
        pro.Name = "Samsung Galaxy S5";
        lstProduct.Add(pro);

        pro = new Product();
        pro.ID = "SSG6";
        pro.Name = "Samsung Galaxy S6";
        lstProduct.Add(pro);

        pro = new Product();
        pro.ID = "SSG8";
        pro.Name = "Samsung Galaxy S8";
        lstProduct.Add(pro);

        return View(lstProduct);
}
```

HomeController.cs

```
@model IEnumerable<Aricle02.Models.Product>
@{
    ViewBag.Title = "Product";
}
<table class="table">
    <tr>
        <th>Product ID</th>
        <th>Product Name</th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(model => item. ID)
            </td>
            <td>
                @Html.DisplayFor(model => item.Name)
            </td>
        </tr>
    }
</table>
```

Index.cshtml

Run… Test


Product - My ASP.NET A
localhost:51324/Home/Index
Application name
Home
About
Contact
Product ID
Product Name
SSG5
Samsung Galaxy S5
SSG6
Samsung Galaxy S6
SSG8
Samsung Galaxy S8
© 2017 - My ASP.NET Application

ASP.NET (C#)

Article 03

In this article, I'll explain an easy but an important concept of how to use – **Model - View - Controller** user in ASP.NET.

```
public ActionResult Index()
{
        return View();
}
public ActionResult Header()
{
        return View();
}
public ActionResult Footer()
{
        return View();
}
```

HomeController.cs

```
public ActionResult Product()
{
        List<Product> lstProduct = new List<Product>();

        Product pro;
        pro = new Product();
        pro.ID = "SSG5";
        pro.Name = "Samsung Galaxy S5";
        lstProduct.Add(pro);

        pro = new Product();
        pro.ID = "SSG6";
        pro.Name = "Samsung Galaxy S6";
        lstProduct.Add(pro);

        pro = new Product();
        pro.ID = "SSG8";
        pro.Name = "Samsung Galaxy S8";
        lstProduct.Add(pro);

        return View(lstProduct);
}
```

HomeController.cs

```
public ActionResult NewProduct()
{
        List<NewProduct> lstProduct = new List<NewProduct>();
        NewProduct pro;
        pro = new NewProduct();
        pro.ID = "IP5";
        pro.Name = "IPhone 5";
        lstProduct.Add(pro);

        pro = new NewProduct();
        pro.ID = "IP6";
        pro.Name = "IPhone 6";
        lstProduct.Add(pro);
        return View(lstProduct);

}
```

HomeController.cs

```
@{
    ViewBag.Title = "Product";
}
@{Html.RenderAction("Header", "Home");}
@{Html.RenderAction("NewProduct", "Home");}
@{Html.RenderAction("Product", "Home");}
@{Html.RenderAction("Footer", "Home");}
```

Index.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - My ASP.NET Application</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <div class="container body-content">
        @RenderBody()
    </div>
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>
```

_Layout.cshtml

Header.cshtml

```
@{
    ViewBag.Title = "Header";
}
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            @Html.ActionLink("Application name", "Index", "Home", null, new
{ @class = "navbar-brand" })
        </div>
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <li>@Html.ActionLink("Home", "Index", "Home")</li>
                <li>@Html.ActionLink("About", "About", "Home")</li>
                <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
            </ul>
        </div>
    </div>
</div>
```

Footer.cshtml

```
@{
    ViewBag.Title = "Footer";
}
<footer>
    <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
</footer>
```

Product.cshtml

```
@model IEnumerable<Article03.Models.Product>
@{
    ViewBag.Title = "Product";
}
<table class="table">
    <tr>
        <th>Product ID</th>
        <th>Product Name</th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(model => item.ID)
            </td>
            <td>
                @Html.DisplayFor(model => item.Name)
            </td>
        </tr>
    }
</table>
```

```
@model IEnumerable<Article03.Models.NewProduct>
@{
    ViewBag.Title = "New Product";
}
<table class="table">
    <tr>
        <th>Product ID</th>
        <th>Product Name</th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(model => item.ID)
            </td>
            <td>
                @Html.DisplayFor(model => item.Name)
            </td>
        </tr>
    }
</table>
```

NewProduct.cshtml

Run …Test

Application name Home About Contact

Product ID	Product Name
IP5	IPhone 5
IP6	IPhone 6

Product ID	Product Name
SSG5	Samsung Galaxy S5
SSG6	Samsung Galaxy S6
SSG8	Samsung Galaxy S8

© 2018 - My ASP.NET Application

ASP.NET (C#)

Article 04

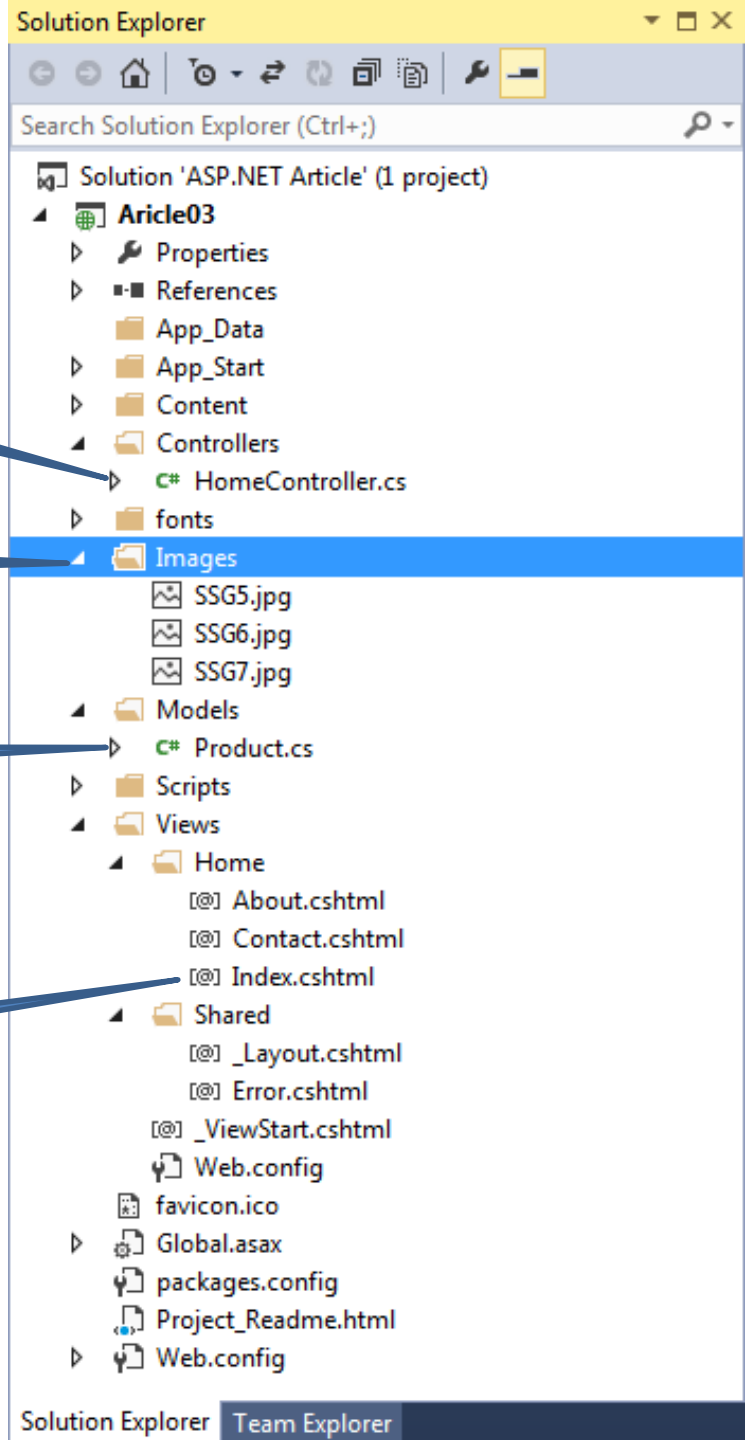
In this article, I'll explain an easy but an important concept of how to use – **Model - View - Controller** user in ASP.NET.

Edit class: HomeController

- Create Folder: Images
- Add Image: SSG5.jpg…

Add new class: Product

Clear all and Edit file: Index.cshtml



```
public class Product

{

        public string ID { get; set; }

        public string Name { get; set; }

        public string Image{ get; set; }

}
```

```
public class HomeController : Controller
{
        public ActionResult Index()
        {
            . . .
        }
}
```

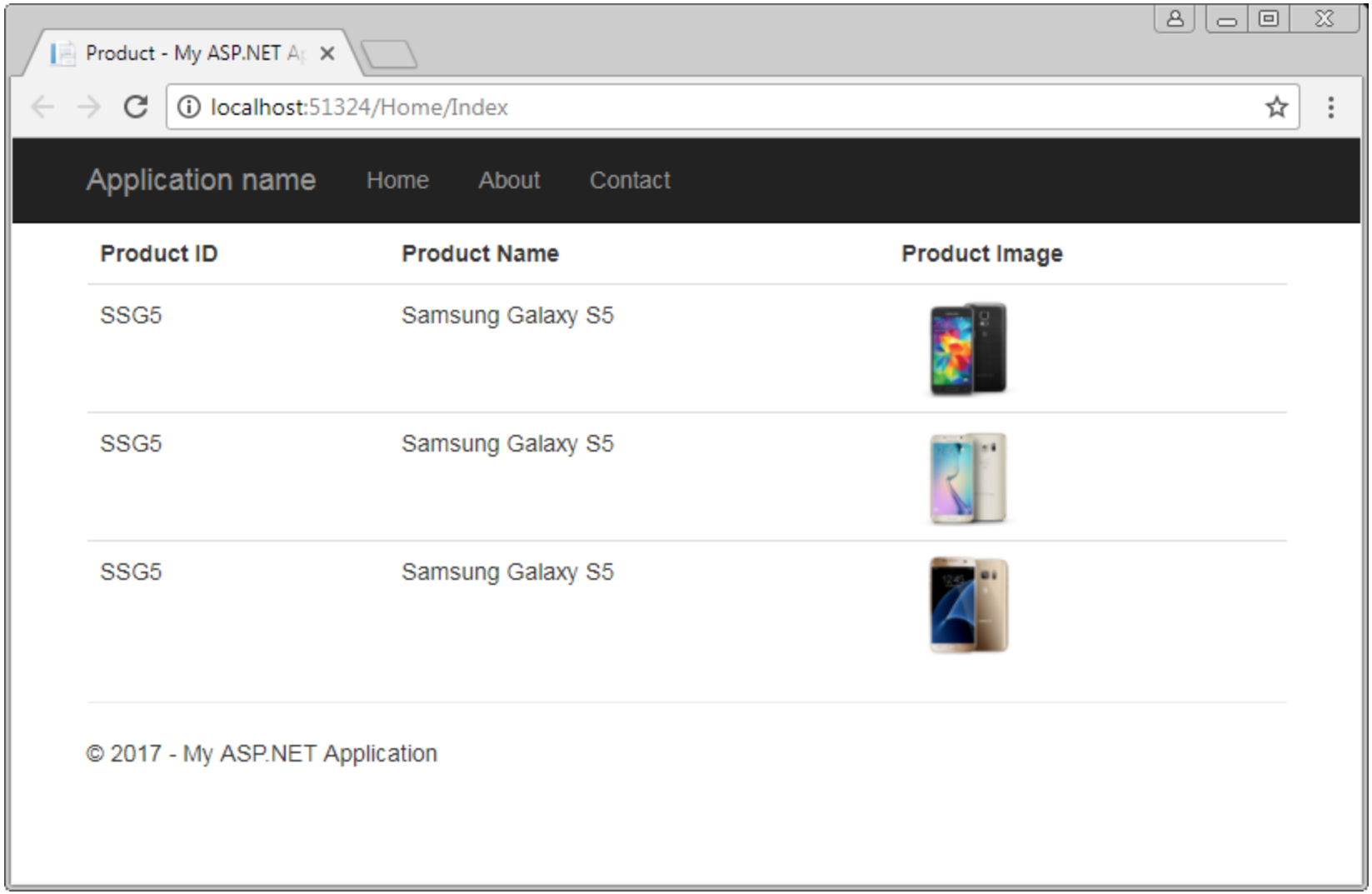
```
public ActionResult Index()
{
        List<Product> lstProduct = new List<Product>();
        Product pro;
        pro = new Product();
        pro.ID = "SSG5";
        pro.Name = "Samsung Galaxy S5";
        pro.Image="~/Images/SSG5.jpg";
        lstProduct.Add(pro);

        pro = new Product();
        pro.ID = "SSG6";
        pro.Name = "Samsung Galaxy S6";
        pro.Image = "~/Images/SSG6.jpg";
        lstProduct.Add(pro);

        pro = new Product();
        pro.ID = "SSG7";
        pro.Name = "Samsung Galaxy S7";
        pro.Image = "~/Images/SSG7.jpg";
        lstProduct.Add(pro);
        return View(lstProduct);
}
```

HomeController.cs

```
@model IEnumerable<Article03.Models.Product>
@{
    ViewBag.Title = "Product";
}
<table class="table">
    <tr>
        <th>Product ID</th>
        <th>Product Name</th>
        <th>Product Image </th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(model => item.ID)
            </td>
            <td>
                @Html.DisplayFor(model => item.Name)
            </td>
            <td>
                <img src="@Url.Content(item.Image)" height="60" width="80"
                                                        alt="Image" />
            </td>
        </tr>
    }
</table>
```

Index.cshtml

Run… Test


Product - My ASP.NET A
localhost:51324/Home/Index
Application name
Home
About
Contact
Product ID
Product Name
Product Image
SSG5
Samsung Galaxy S5
SSG5
Samsung Galaxy S5
SSG5
Samsung Galaxy S5
© 2017 - My ASP.NET Application

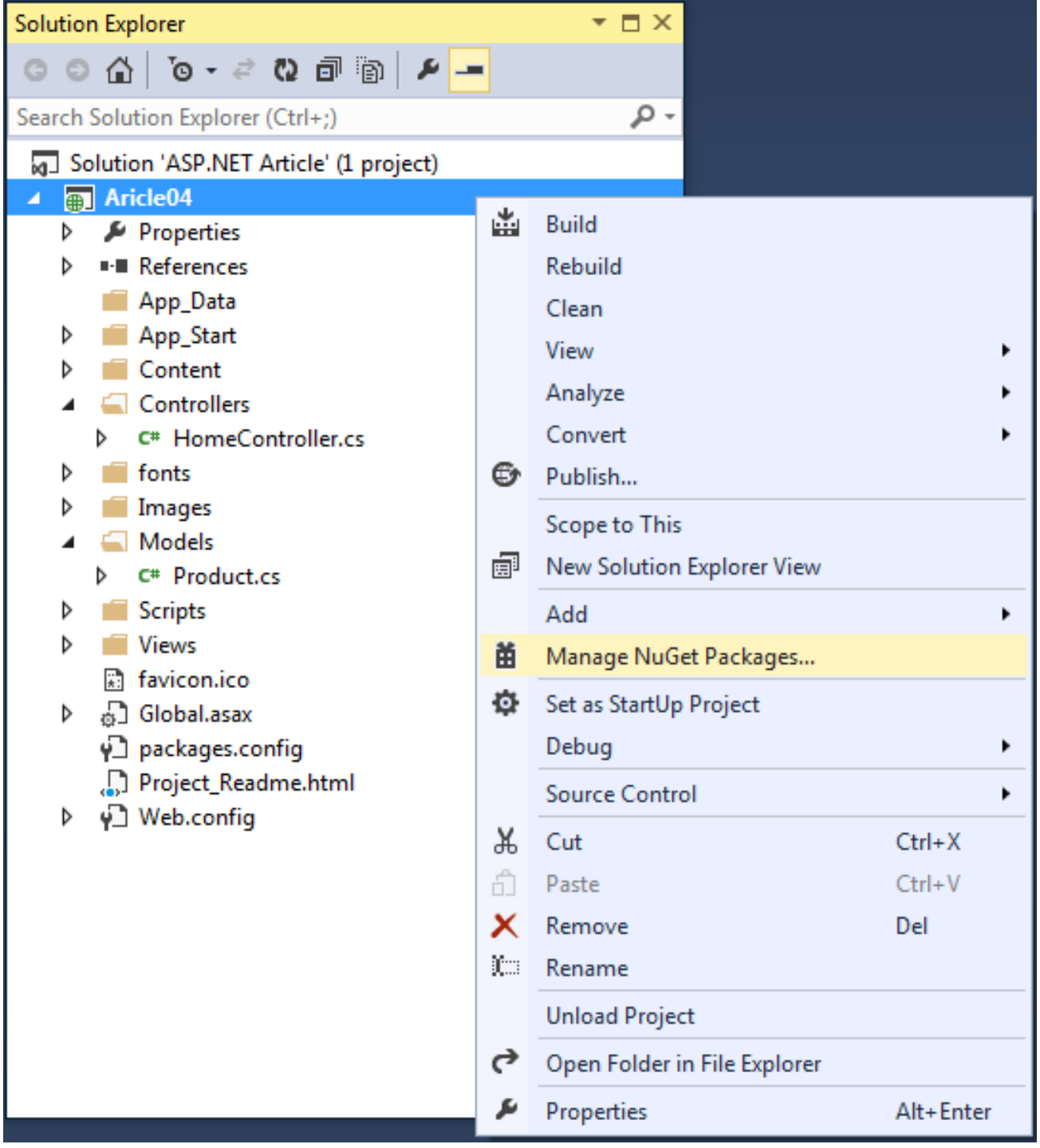
ASP.NET

PagedList

ASP.NET (C#)

Article 05

In this article, I'll explain an easy but an important concept of how to use **PagedList** in ASP.NET (MVC).


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ASP.NET Article' (1 project)
Aricle04
Properties
References
App_Data
App_Start
Content
Controllers
HomeController.cs
fonts
Images
Models
Product.cs
Scripts
Views
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config
Build
Rebuild
Clean
View
Analyze
Convert
Publish...
Scope to This
New Solution Explorer View
Add
Manage NuGet Packages...
Set as StartUp Project
Debug
Source Control
Cut Ctrl+X
Paste Ctrl+V
Remove Del
Rename
Unload Project
Open Folder in File Explorer
Properties Alt+Enter

Aricle04 - Manage NuGet Packages

Installed packages

Online

All

nuget.org

Microsoft and .NET

Search Results

Updates

Stable Only | Sort by: Relevance | PagedList

PagedList.Mvc
Asp.Net MVC HtmlHelper method for generating paging control for use with Page...
Install

PagedList
PagedList makes it easier for .Net developers to write paging code. It allows you to take any IEnumerable(T) and by specify...

PagedList for ASP.NET MVC (bootstrap)
http://www.calabonga.net/blog/post/53

AutoMapperPagedList
Adds an implementation of PagedList that uses AutoMapper to emit ViewModels

PagedListLite
Lightweight and serializable PagedList implementation for .NET (IQuryable<T>, IEnumerable<T>, ICollection<T>)

Caseiro.Mvc.PagedList
A paged list helper for ASP.NET MVC with filter and order functionalities

ASP.NET Core PagedList Async Extension Library
This package provides extension method to create PagedList using EntityFramework async method.

1 2 3 4

Created by: Troy Goo
Id: PagedList.Mvc
Version: 4.5.0.0
Published: 30/09/2013
Downloads: 1366516
License
View License
Project Information
Report Abuse
Description:
Asp.Net MVC HtmlHelper method for generating paging control for use with PagedList library.
Tags: paging page infinitescroll ajax mvc
Dependencies:
PagedList (≥ 1.17)
Each item above may have sub-dependencies subject to additional license agreements.

Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.

Settings | Close

1. Online

2. PagedList

3. Click Install (PagedList.Mvc)

4. Click Install (PagedList)

HayMuaSiModel - Manage NuGet Packages

Installed packages

All

Online

Updates

Sort by: Name: Ascending

Search Installed packages (Ctrl+E)

Modernizr
Modernizr is a small and simple JavaScript library that helps you take advantage of emerging web technologies (CSS3, HT...

PagedList.Mvc
Asp.Net MVC HtmlHelper method for generating paging control for use with Page...
Uninstall

PagedList
PagedList makes it easier for .Net developers to write paging code. It allows you to take any IEnumerable(T) and by specify...

Respond JS
A fast & lightweight polyfill for min/max-width CSS3 Media Queries (for IE 6-8, and more)

WebGrease
Web Grease is a suite of tools for optimizing javascript, css files and images.

Created by: Troy Goode
Id: PagedList.Mvc
Version: 4.5.0.0
License
View License
Project Information
Description:
Asp.Net MVC HtmlHelper method for generating paging control for use with PagedList library.
Tags: paging pager page infinitescroll ajax mvc
Dependencies:
PagedList (≥ 1.17)

Each item above may have sub-dependencies subject to additional license agreements.

Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.

1 2 3

Settings

Close

Search Installed packages (Ctrl+E)

le JavaScript library that helps
ng web technologies (CSS3, HT...

thod for generating paging
library.

PagedList
PagedList makes it easier for .Net developers to write paging code. It allows you to take a...
Uninstall

Respond JS
A fast & lightweight polyfill for min/max-width CSS3 Media Queries (for IE 6-8, and more)

WebGrease
Web Grease is a suite of tools for optimizing javascript, css files and images.

Created by: Troy Goode
Id: PagedList
Version: 1.17.0.0
License
View License
Project Information
Description:
PagedList makes it easier for .Net developers to write paging code. It allows you to take any IEnumerable(T) and by specifying the page size and desired page index, select only a subset of that list. PagedList also provides properties that are useful when building UI paging controls.
Tags: paging pager page infinitescroll ajax mvc
Dependencies:
No Dependencies

Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.

1 2 3

Settings

Close

Model

```
public class Product
{
        public string ID { get; set; }
        public string Name { get; set; }
        public string Image{ get; set; }
        public int Price{ get; set; }
}
```

```
public class HomeController : Controller
{
        public ActionResult Index()
        {
                . . .
        }
}
```

```
public class HomeController : Controller {
    public List<Product> CreateData() {
        List<Product> lstProduct = new List<Product>();
        Product pro;
        pro = new Product();
        pro.ID = "SSG5";
        pro.Name = "Samsung Galaxy S5";
        pro.Image="~/Images/SSG5.jpg";
        pro.Price = 1000000;
        lstProduct.Add(pro);

        pro= new Product();
        pro.ID = "SSG6";
        pro.Name = "Samsung Galaxy S6";
        pro.Image = "~/Images/SSG6.jpg";
        pro.Price = 22000000;
        lstProduct.Add(pro);

        pro = new Product();
        pro.ID = "SSG7";
        pro.Name = "Samsung Galaxy S7";
        pro.Image = "~/Images/SSG7.jpg";
        pro.Price = 15000000;
        lstProduct.Add(pro);
        // Add >20 Product
        return lstProduct;
    }
    public ActionResult Index() {
        . . .
    }
```

HomeController.cs

HomeController.cs

```
using . . .
using PagedList;

public class HomeController : Controller {
    public List<Product> CreateData() { . . .}

        public ActionResult Index(int? page)
        {
          List<Product> listProduct = CreateData();
          int pageSize = 5;
          int pageNumber = (page ?? 1);
          return View(listProduct.ToPagedList(pageNumber, pageSize));
        }
}
```

```
model PagedList.IPagedList<Aricle04.Models.Product>
@using PagedList.Mvc;
@{
    ViewBag.Title = "Product";
}

<table class="table">

    . . .

</table>

<div class="pagelist" style="padding: 0px 0px;">
  @Html.PagedListPager(Model, page => Url.Action("Index", new { page }))
</div>
```

Same Article03

Product - My ASP.NET A

localhost:51324

Application name Home About Contact

Product ID	Product Name	Product Image
SSG5	Samsung Galaxy S5	
SSG6	Samsung Galaxy S6	
SSG7	Samsung Galaxy S7	
SSG8	Samsung Galaxy S8	
SSG9	Samsung Galaxy S9	

1 2 »

ASP.NET

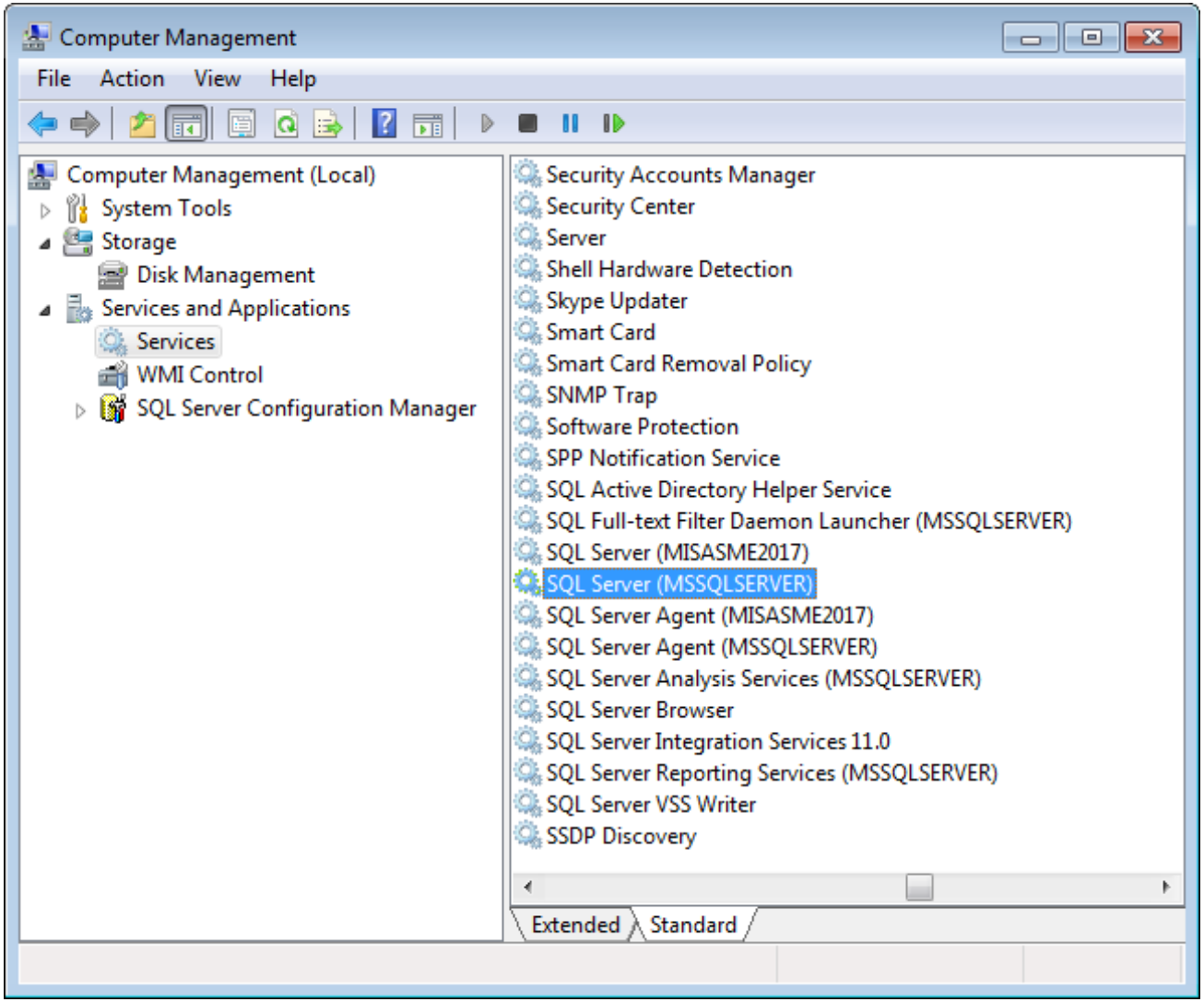
Entity Framework (EF)

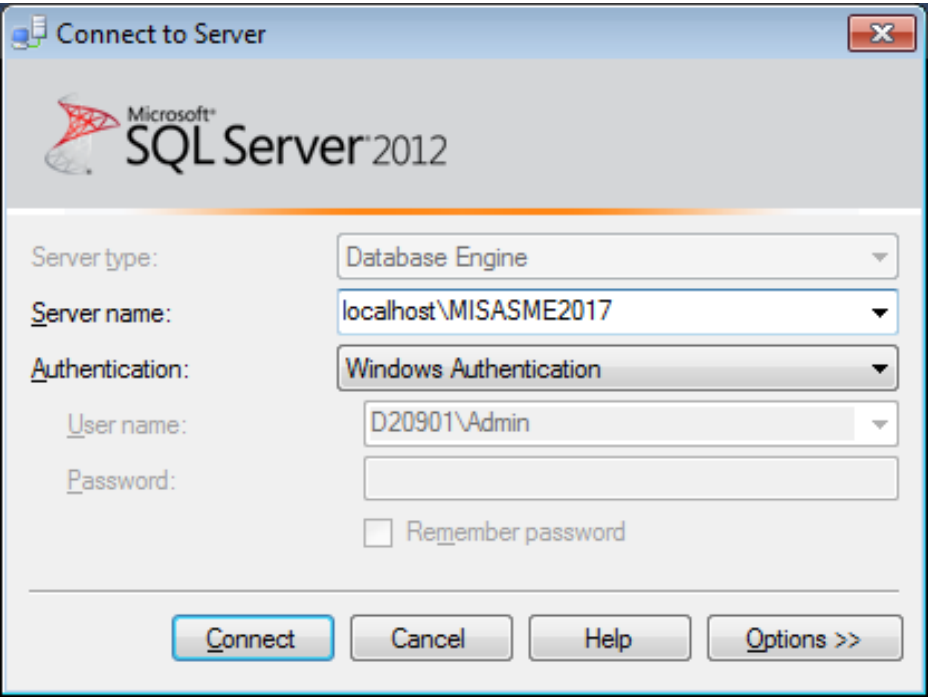
ASP.NET (C#)

Article 06

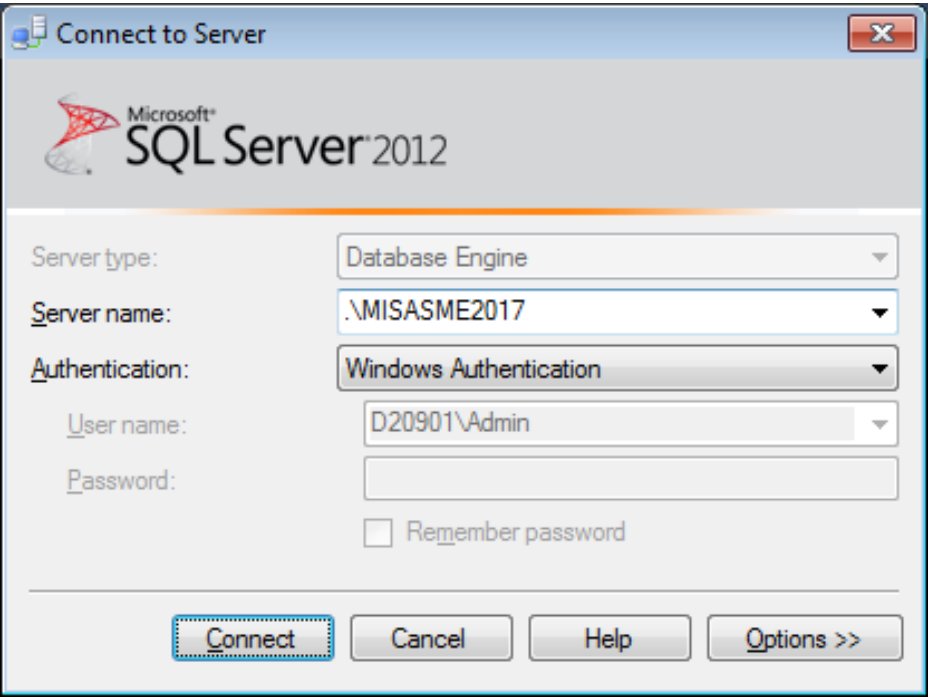
In this article, I'll explain an easy but an important concept of how to use **sa** user in SQL Server 201x.

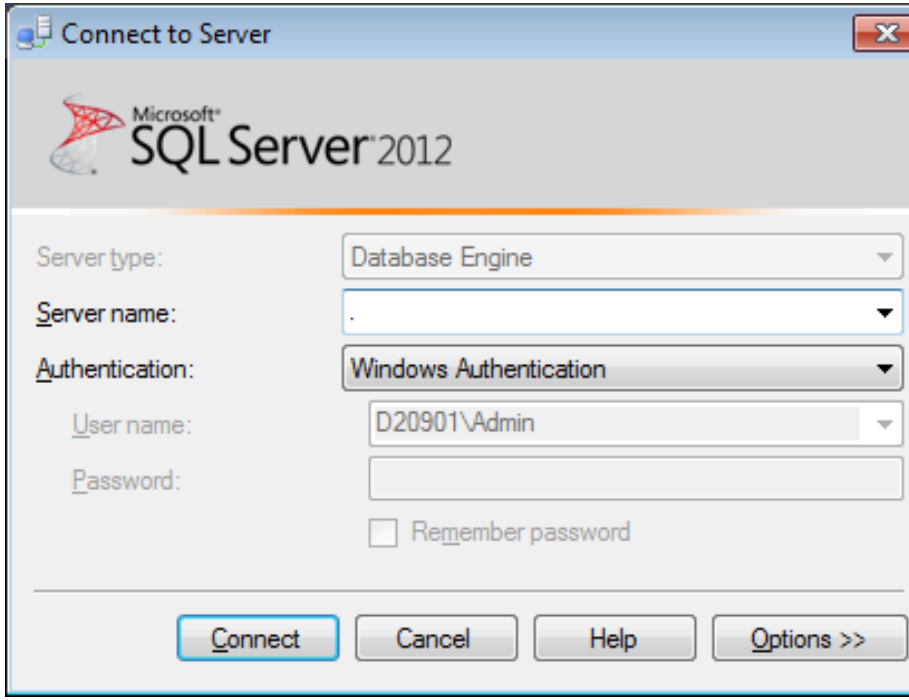
- **SQL Server 201x**
  - ✓ user: sa
  - ✓ password: sa
  - ✓ Server Authentication: SQL Server and Windows mode


Computer Management
File
Action
View
Help
Computer Management (Local)
System Tools
Storage
Disk Management
Services and Applications
Services
WMI Control
SQL Server Configuration Manager
Security Accounts Manager
Security Center
Server
Shell Hardware Detection
Skype Updater
Smart Card
Smart Card Removal Policy
SNMP Trap
Software Protection
SPP Notification Service
SQL Active Directory Helper Service
SQL Full-text Filter Daemon Launcher (MSSQLSERVER)
SQL Server (MISASME2017)
SQL Server (MSSQLSERVER)
SQL Server Agent (MISASME2017)
SQL Server Agent (MSSQLSERVER)
SQL Server Analysis Services (MSSQLSERVER)
SQL Server Browser
SQL Server Integration Services 11.0
SQL Server Reporting Services (MSSQLSERVER)
SQL Server VSS Writer
SSDP Discovery
Extended
Standard


Connect to Server
Microsoft SQL Server 2012
Server type:
Database Engine
Server name:
localhost\MISASME2017
Authentication:
Windows Authentication
User name:
D20901\Admin
Password:
Remember password
Connect
Cancel
Help
Options >>


Connect to Server
Microsoft SQL Server 2012
Server type:
Database Engine
Server name:
localhost
Authentication:
Windows Authentication
User name:
D20901\Admin
Password:
Remember password
Connect
Cancel
Help
Options >>


Connect to Server
Microsoft SQL Server 2012
Server type:
Database Engine
Server name:
.\MISASME2017
Authentication:
Windows Authentication
User name:
D20901\Admin
Password:
Remember password
Connect
Cancel
Help
Options >>


Connect to Server
Microsoft SQL Server 2012
Server type:
Database Engine
Server name:
.
Authentication:
Windows Authentication
User name:
D20901\Admin
Password:
Remember password
Connect
Cancel
Help
Options >>

Object Explorer

Connect

- localhost (SQL Server 11.0.2100 - D20901\Admin)
  - Databases
  - Security
    - Logins
      - ##MS_PolicyEventProcessingLogin##
      - ##MS_PolicyTsqlExecutionLogin##
      - NT AUTHORITY\SYSTEM
      - NT Service\MSSQLSERVER
      - NT SERVICE\ReportServer
      - NT SERVICE\SQLSERVERAGENT
      - NT SERVICE\SQLWriter
      - NT SERVICE\Winmgmt
      - sa
      - USER-PC\Admin
      - USER-PC\Administrator
    - Server Roles
    - Credentials
    - Cryptographic Providers
    - Audits
    - Server Audit Specifications
  - Server Objects
  - Replication
  - AlwaysOn High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabled)

Right click -> Properties

Login Properties - sa

Select a page
- General
- Server Roles
- User Mapping
- Status

Script Help

Settings

Permission to connect to database engine:
- (•) Grant
- ( ) Deny

Login:
- (•) Enabled
- ( ) Disabled

Status

SQL Server authentication:
- [ ] Login is locked out

Connection

Server:
DESKTOP-EDLEDHP\KORO

Connection:
DESKTOP-EDLEDHP\KORO

View connection properties

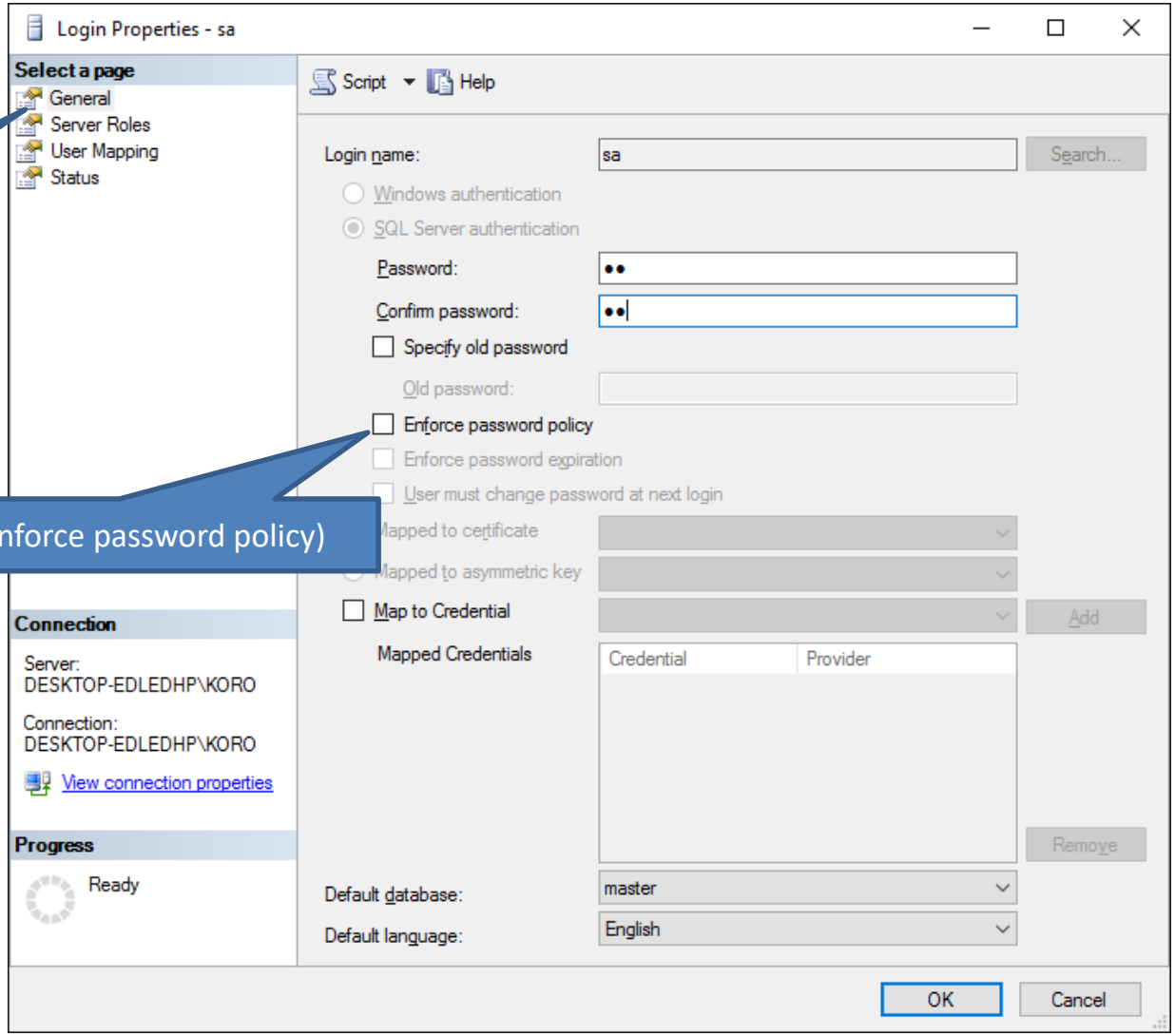
Progress

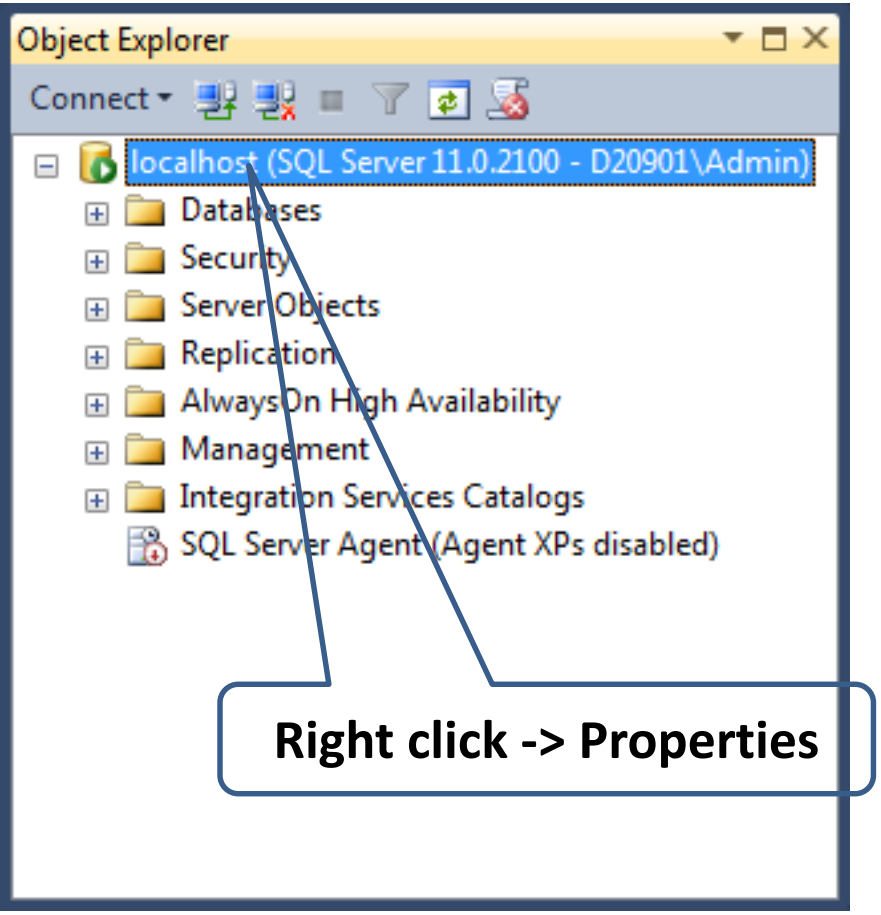
Ready

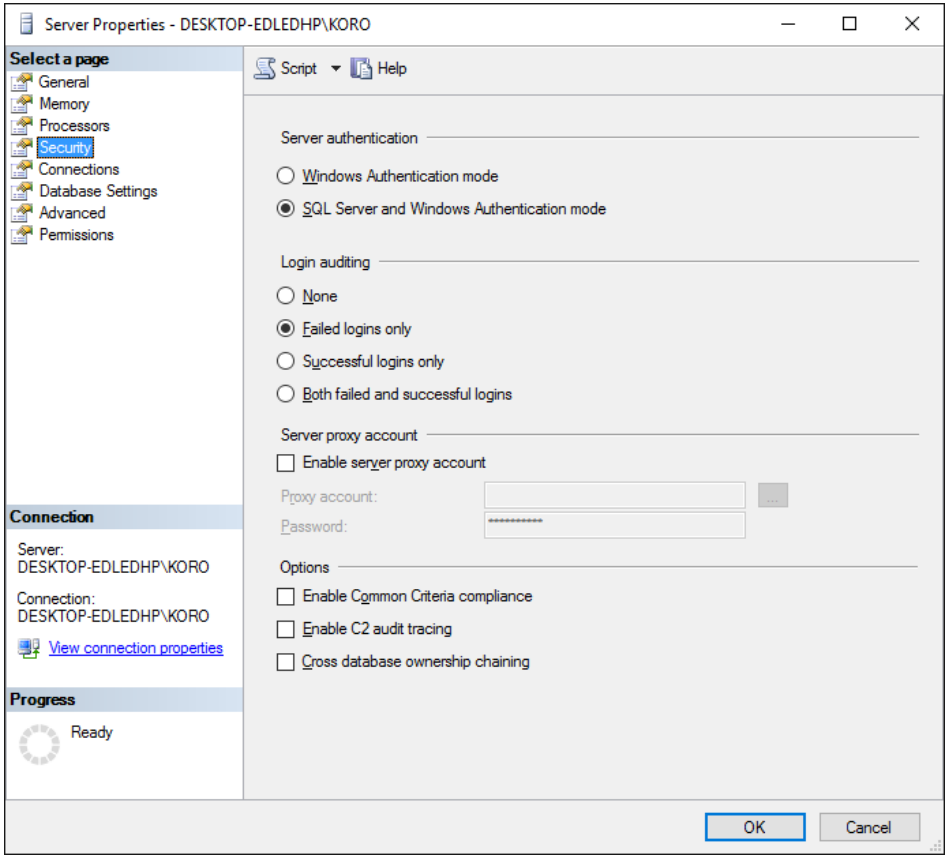
OK Cancel

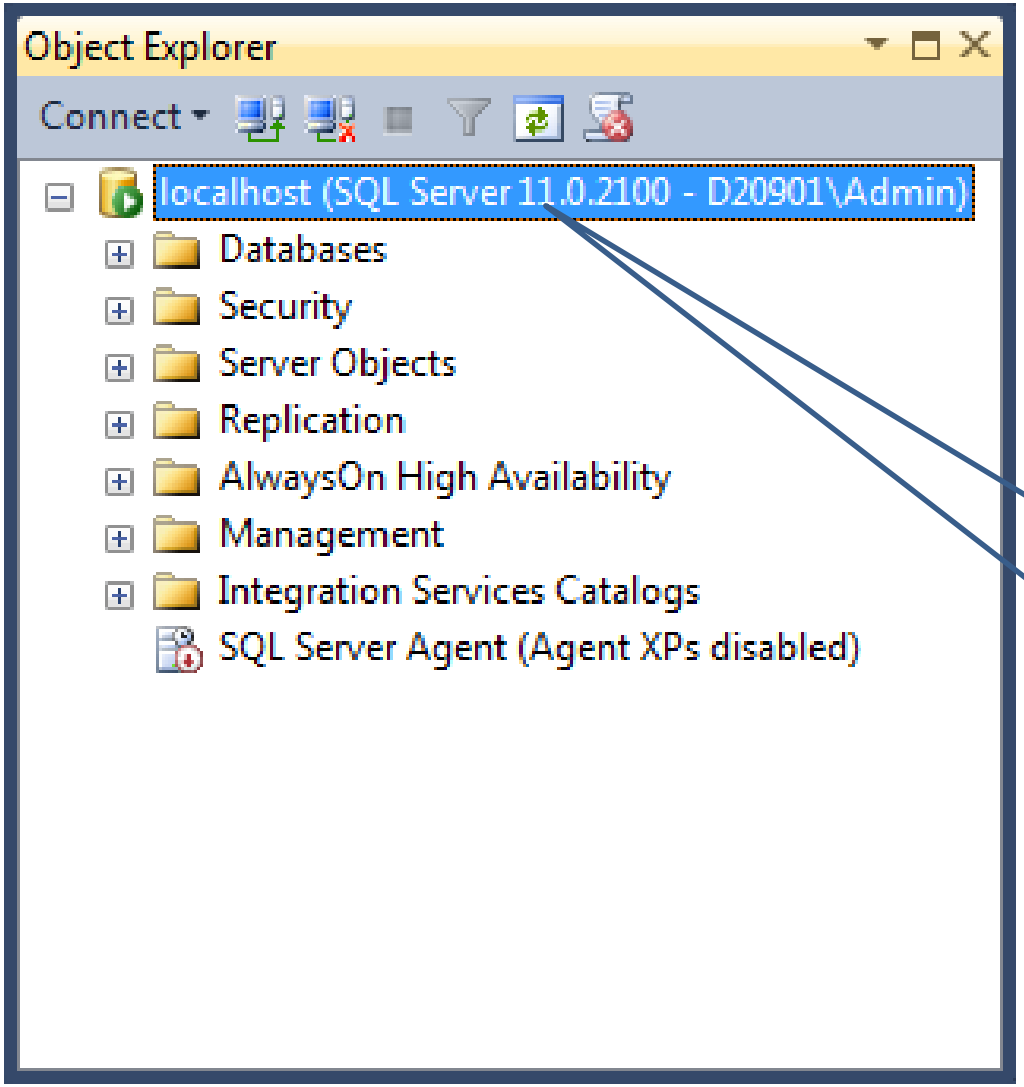
1. General

2. Uncheck (Enforce password policy)



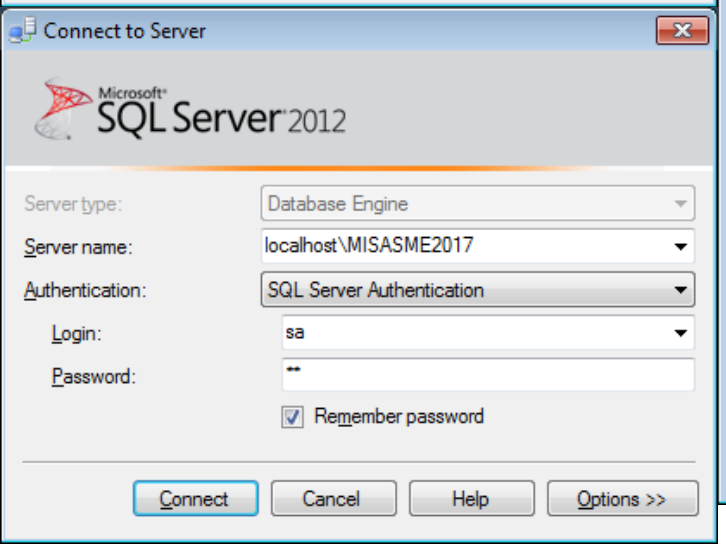

Object Explorer
Connect
localhost (SQL Server 11.0.2100 - D20901\Admin)
Databases
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
Right click -> Properties

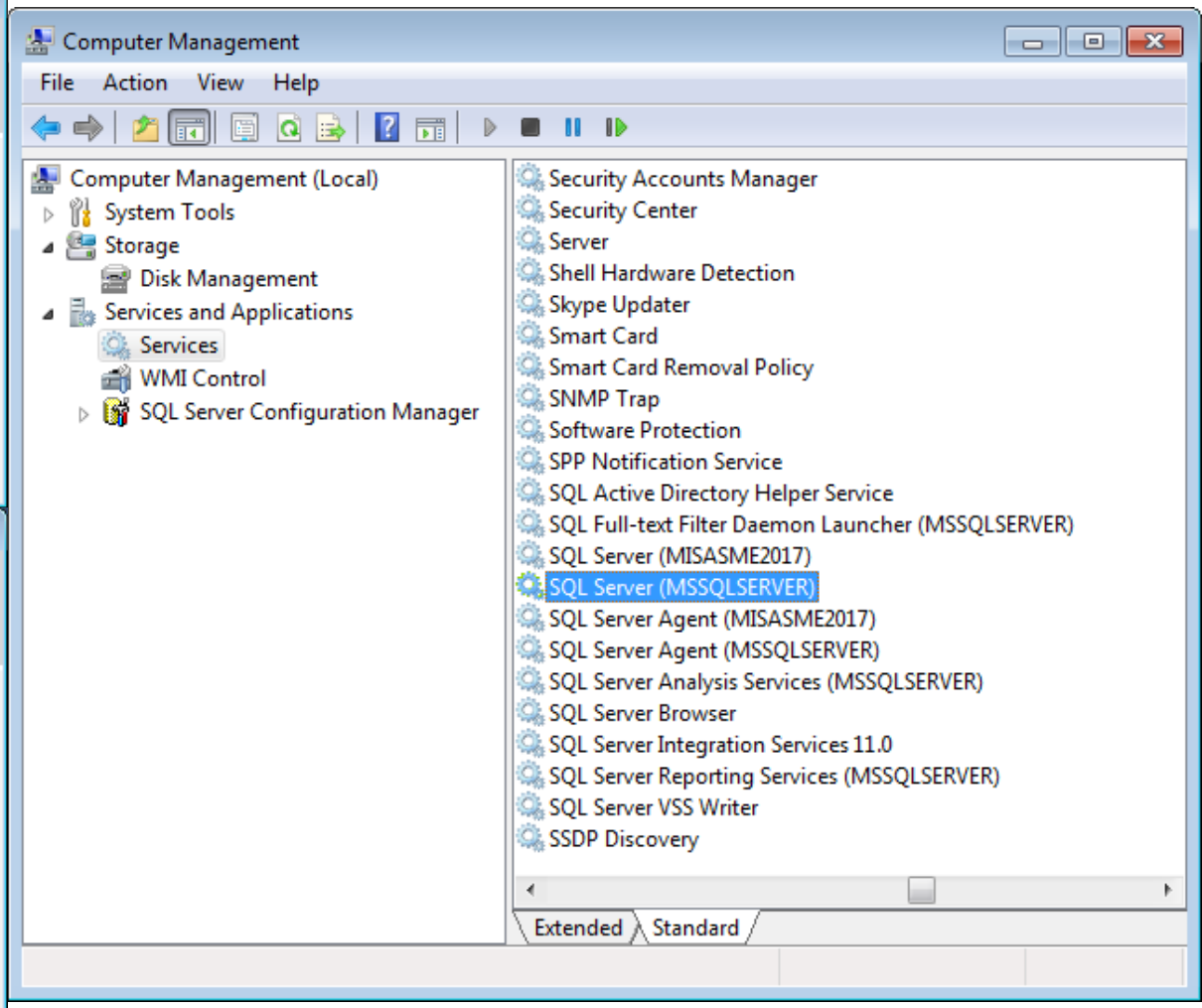

Server Properties - DESKTOP-EDLEDHP\KORO
Select a page
General
Memory
Processors
Security
Connections
Database Settings
Advanced
Permissions
Script
Help
Server authentication
Windows Authentication mode
SQL Server and Windows Authentication mode
Login auditing
None
Failed logins only
Successful logins only
Both failed and successful logins
Server proxy account
Enable server proxy account
Proxy account:
Password:
Options
Enable Common Criteria compliance
Enable C2 audit tracing
Cross database ownership chaining
Connection
Server:
DESKTOP-EDLEDHP\KORO
Connection:
DESKTOP-EDLEDHP\KORO
View connection properties
Progress
Ready
OK
Cancel



Right click -> Restart


Connect to Server
Microsoft SQL Server 2012
Server type:
Database Engine
Server name:
localhost
Authentication:
SQL Server Authentication
Login:
sa
Password:
Remember password
Connect
Cancel
Help
Options >>

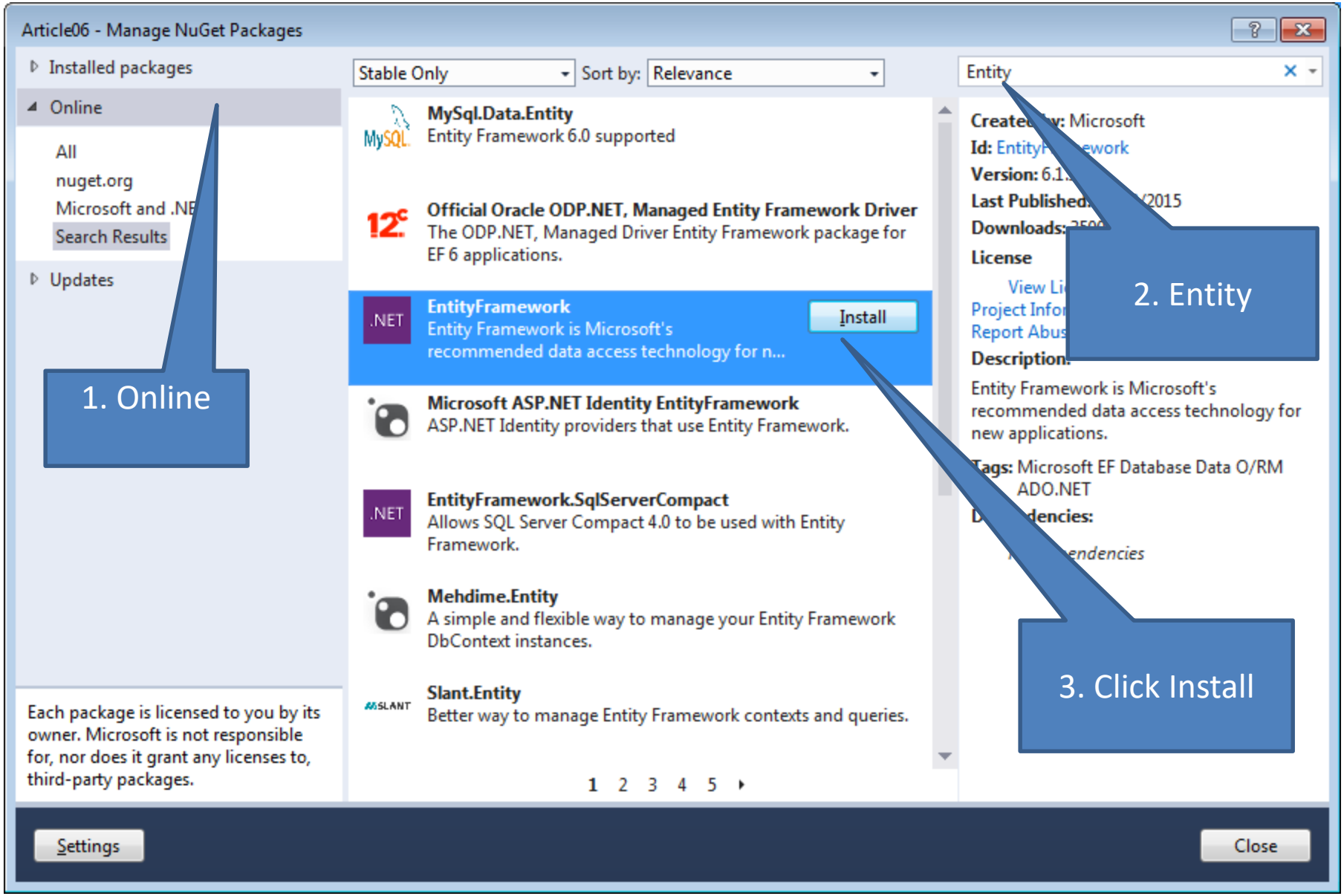

Connect to Server
Microsoft SQL Server 2012
Server type:
Database Engine
Server name:
localhost\MISASME2017
Authentication:
SQL Server Authentication
Login:
sa
Password:
Remember password
Connect
Cancel
Help
Options >>

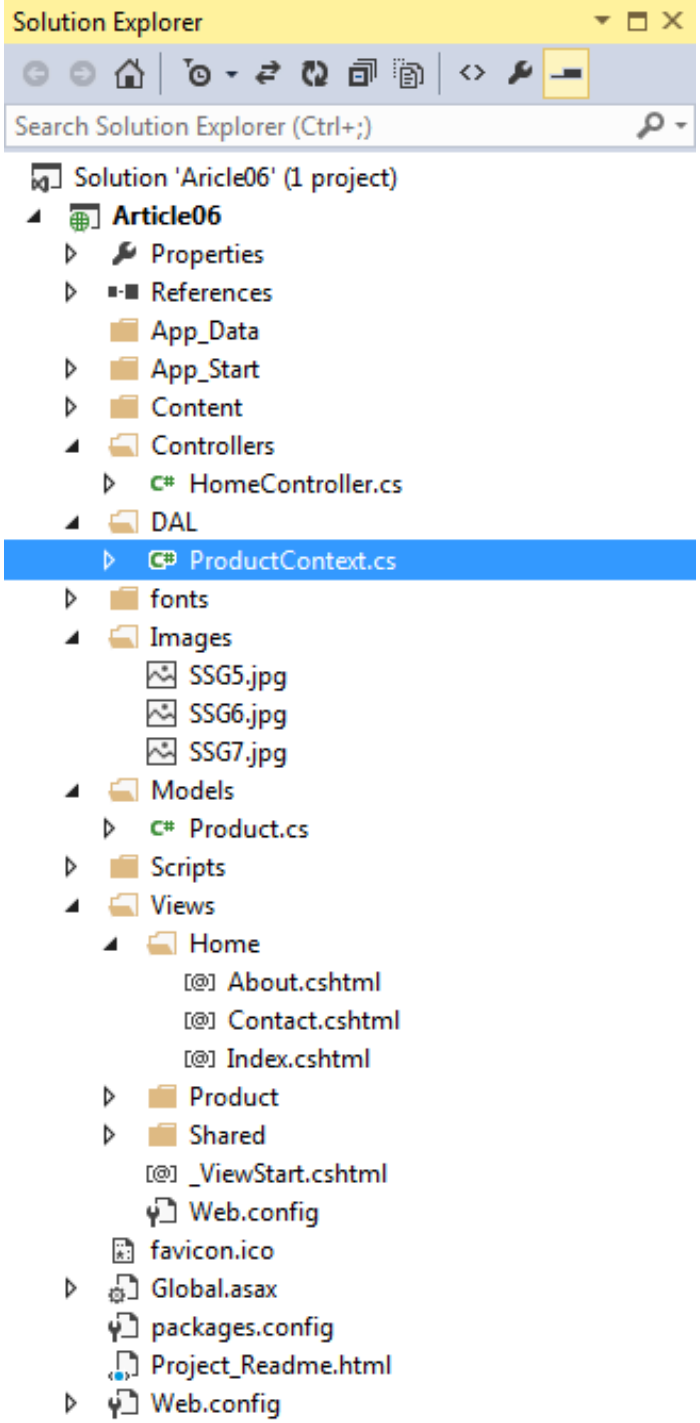

Computer Management
File
Action
View
Help
Computer Management (Local)
System Tools
Storage
Disk Management
Services and Applications
Services
WMI Control
SQL Server Configuration Manager
Security Accounts Manager
Security Center
Server
Shell Hardware Detection
Skype Updater
Smart Card
Smart Card Removal Policy
SNMP Trap
Software Protection
SPP Notification Service
SQL Active Directory Helper Service
SQL Full-text Filter Daemon Launcher (MSSQLSERVER)
SQL Server (MISASME2017)
SQL Server (MSSQLSERVER)
SQL Server Agent (MISASME2017)
SQL Server Agent (MSSQLSERVER)
SQL Server Analysis Services (MSSQLSERVER)
SQL Server Browser
SQL Server Integration Services 11.0
SQL Server Reporting Services (MSSQLSERVER)
SQL Server VSS Writer
SSDP Discovery
Extended
Standard

ASP.NET (C#)

Article 07

In this article, I'll explain an easy but an important concept of how to Read data.


Article06 - Manage NuGet Packages
Installed packages
Online
All
nuget.org
Microsoft and .NE
Search Results
Updates
Stable Only
Sort by: Relevance
Entity
MySql.Data.Entity
Entity Framework 6.0 supported
Official Oracle ODP.NET, Managed Entity Framework Driver
The ODP.NET, Managed Driver Entity Framework package for EF 6 applications.
EntityFramework
Entity Framework is Microsoft's recommended data access technology for n...
Install
Microsoft ASP.NET Identity EntityFramework
ASP.NET Identity providers that use Entity Framework.
EntityFramework.SqlServerCompact
Allows SQL Server Compact 4.0 to be used with Entity Framework.
Mehdime.Entity
A simple and flexible way to manage your Entity Framework DbContext instances.
Slant.Entity
Better way to manage Entity Framework contexts and queries.
1 2 3 4 5
Id:
Version:
Last Published:
Downloads:
License
Project Infor
Report Abus
Description:
Entity Framework is Microsoft's recommended data access technology for new applications.
Tags: Microsoft EF Database Data O/RM ADO.NET
Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Settings
Close
1. Online
2. Entity
3. Click Install


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Aricle06' (1 project)
Article06
Properties
References
App_Data
App_Start
Content
Controllers
HomeController.cs
DAL
ProductContext.cs
fonts
Images
SSG5.jpg
SSG6.jpg
SSG7.jpg
Models
Product.cs
Scripts
Views
Home
About.cshtml
Contact.cshtml
Index.cshtml
Product
Shared
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config

Product.CS

```
public class Product
{
        public string ID { get; set; }
        public string Name { get; set; }
        public string Image{ get; set; }
        public string Detail{ get; set; }
        public int Price{ get; set; }
}
```

ProductContext.cs

```
namespace Article07.DAL
{
    public class ProductContext : DbContext
    {
        public ProductContext() : base("name=ProductContext")
        {
        }
        public System.Data.Entity.DbSet<Product> Products { get; set; }
    }
}
```

Edit *Web.config* the following changes

```
<configuration>
 . . .
 <connectionStrings>
    <add name= "ProductContext"
providerName="System.Data.SqlClient" connectionString="Data
Source=(local);Initial Catalog=Sale;Integrated Security=False;
User Id=sa;Password=sa; MultipleActiveResultSets=True" />
  </connectionStrings>
. . .
</configuration>
```

HomeController.cs

```
. . .
using Article07.DAL;
public class HomeController : Controller
{
        private ProductContext db = new ProductContext();

        public ActionResult Index()
        {
                return View(db.Products.ToList());
        }
}
```

Index.cshtml

```
@model IEnumerable<Article07.Models.Product>
@{
    ViewBag.Title = "Product";
}
<table class="table">
    <tr>
        <th>Product ID</th>
        <th>Product Name</th>
        <th>Product Image </th>
    </tr>
    @foreach (var item in Model){
    <tr>
        <td>
            @Html.DisplayFor(model => item.ID)
        </td>
        <td>
            @Html.DisplayFor(model => item.Name)
        </td>
        <td>
            <img src="@Url.Content(item.Image)" height="60" width="80"
            alt="Image" />
        </td>
    </tr>
    }
</table>
```

Product - My ASP.NET A

localhost:49393

Application name

Product ID	Product Name	Product Image

© 2017 - My ASP.NET Application

Object Explorer

Connect

- . (SQL Server 11.0.2100 - sa)
  - Databases
    - System Databases
    - Database Snapshots
    - ReportServer
    - ReportServerTempDB
    - Sale
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - dbo.__MigrationHistory
        - dbo.Products
      - Views
      - Synonyms
      - Programmability
      - Service Broker
      - Storage
      - Security
  - Security
  - Server Objects
  - Replication
  - AlwaysOn High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabled)

```
use Sale
go
    insert into Products values ('SSG5',N'Samsung Galaxy
 S5','~/Images/SSG5.jpg','Độ phân giải màn hình:1080x1920
 pixels, Tốc độ CPU:Quad-core 1.9 GHz Cortex-A15',5800000)

    insert into Products values ('SSG6',N'Samsung Galaxy
 S6','~/Images/SSG6.jpg ','Độ phân giải màn hình:1080x1920
 pixels, Tốc độ CPU:Quad-core 2.5 GHz Cortex-A16', 8000000)

    insert into Products values ('SSG7',N'Samsung Galaxy
 S7','~/Images/SSG7.jpg ','Độ phân giải màn hình:1080x1920
 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17', 1500000)
```

SQLQuery2.sql - (local).Sale (sa (56))* - Microsoft SQL Server Management Studio (Administrator)

File Edit View Query Project Debug Tools Window Help

New Query | Sale | Execute | Debug

Object Explorer

Connect

- . (SQL Server 11.0.2100 - sa)
  - Databases
    - System Databases
    - Database Snapshots
    - haymuasi
    - ReportServer
    - ReportServerTempDB
    - Sale
  - Security
  - Server Objects
  - Replication
  - AlwaysOn High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabl

SQLQuery2.sql - (local).Sale (sa (56))* | SQLQuery1.sql - (lo...).haymuasi (sa (52))*

```
use Sale
go
insert into Products values ('SSG5','Samsung Galaxy S5','~/Images/SSG5.jpg',N'Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Qu

insert into Products values ('SSG6','Samsung Galaxy S6','~/Images/SSG6.jpg ',N'Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Q

insert into Products values ('SSG7','Samsung Galaxy S7','~/Images/SSG7.jpg ',N'Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Q

select * from Products
```

100 %

Results | Messages

	ID	Name	Image	Detail	Price
1	SSG5	Samsung Galaxy S5	~/Images/SSG5.jpg	Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 1.9 GHz Cortex-A15	5800000
2	SSG6	Samsung Galaxy S6	~/Images/SSG6.jpg	Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.5 GHz Cortex-A16	8000000
3	SSG7	Samsung Galaxy S7	~/Images/SSG7.jpg	Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17	1500000

Query executed successfully. | (local) (11.0 RTM) | sa (56) | Sale | 00:00:00 | 3 rows

Ready | Ln 1 | Col 1

Properties

Product - My ASP.NET A

localhost:49393

Application name

Product ID	Product Name	Product Image
SSG5	Samsung Galaxy S5	
SSG6	Samsung Galaxy S6	
SSG7	Samsung Galaxy S7	

ASP.NET (C#)

Article 08

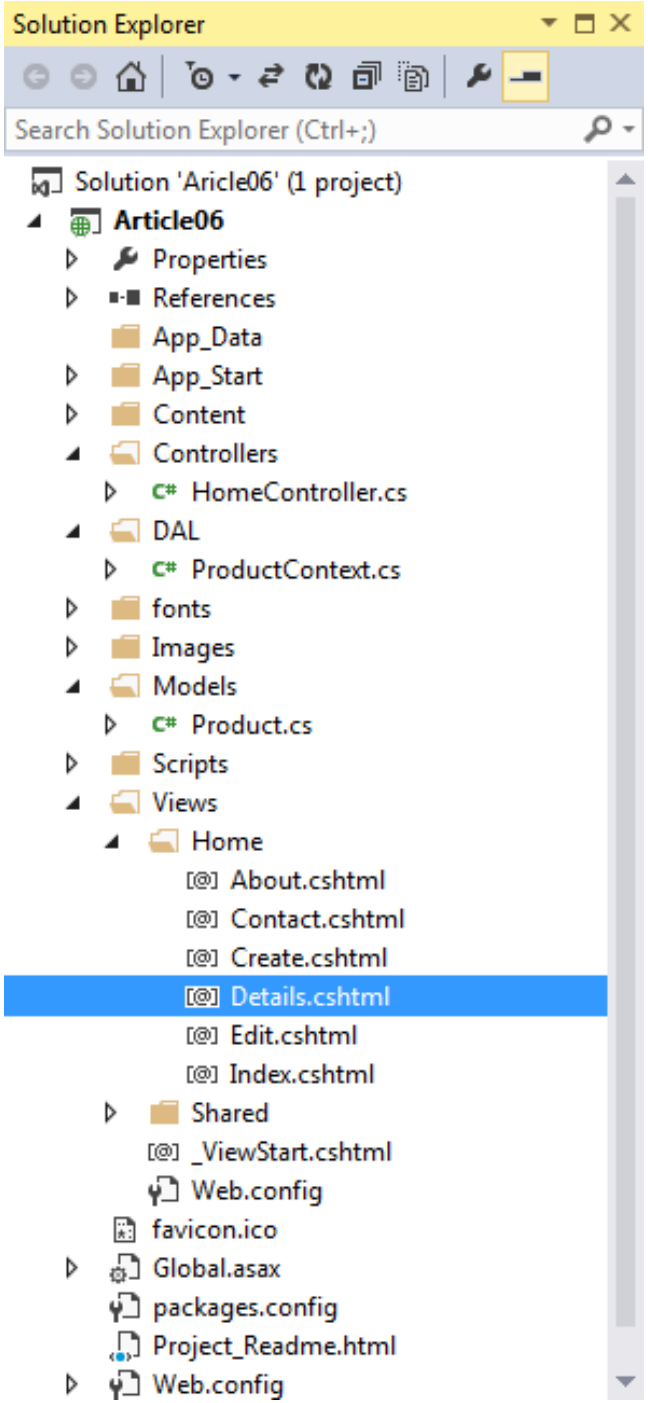
In this article, I'll explain an easy but an important concept of how to Detail data.

View Index.cshtml

```
...
@foreach (var item in Model) {
<tr>
    <td>
        @Html.DisplayFor(model => item.ID)
    </td>
    <td>
        @Html.DisplayFor(model => item.Name)
    </td>
    <td>
        @Html.DisplayFor(model => item.Price)
    </td>
    <td>
        <img src="@Url.Content(item.Image)" height="60" width="80" alt="Image" />
    </td>
    <td>
        @Html.ActionLink("Details", "Details", new { id = item.ID })
    </td>
</tr>
}
</table>
```

HomeController

```
private ProductEntity db = new ProductEntity();
public ActionResult Index()
{
        . . .
}
. . .
public ActionResult Details(string id)
{
        if (id == null)
        {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Product product = db.Products.Find(id);
        if (product == null)
        {
                return HttpNotFound();
        }
        return View(product);
}
```


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Aricle06' (1 project)
Article06
Properties
References
App_Data
App_Start
Content
Controllers
HomeController.cs
DAL
ProductContext.cs
fonts
Images
Models
Product.cs
Scripts
Views
Home
About.cshtml
Contact.cshtml
Create.cshtml
Details.cshtml
Edit.cshtml
Index.cshtml
Shared
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config

View Detail.cshtml

```
@model Article08.Models.Product
@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Details</h2>

<div>
    <h4>Product</h4>
<hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Image)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Image)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Name)
        </dt>
```

View Detail.cshtml

```
        <dd>
            @Html.DisplayFor(model => model.Name)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Price)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Price)
        </dd>
        <dd>
            @Html.DisplayFor(model => model.Detail)
        </dd>

    </dl>
</div>
<p>
        @Html.ActionLink("Back to List", "Index")
</p>
```

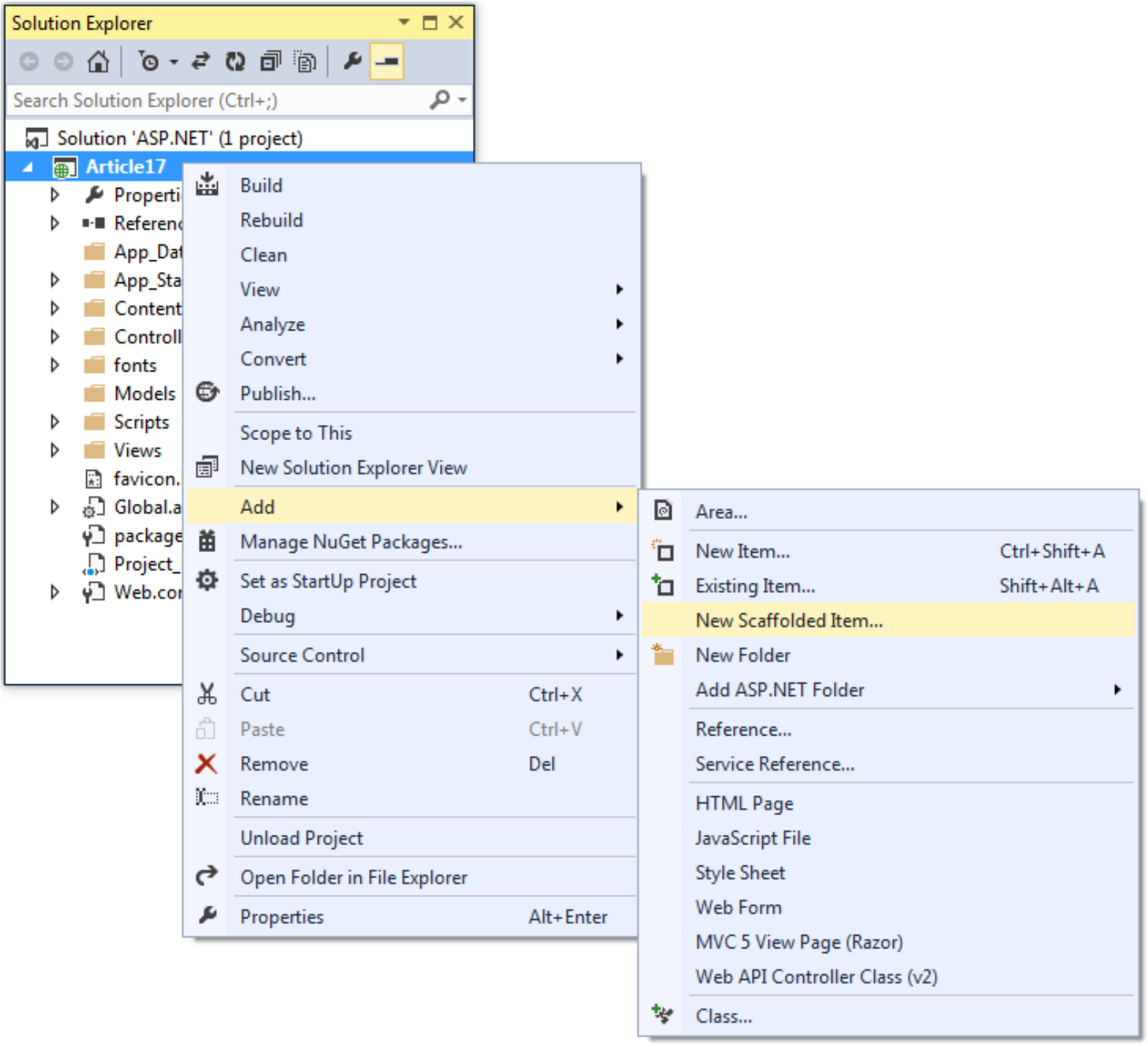
View Index.cshtml

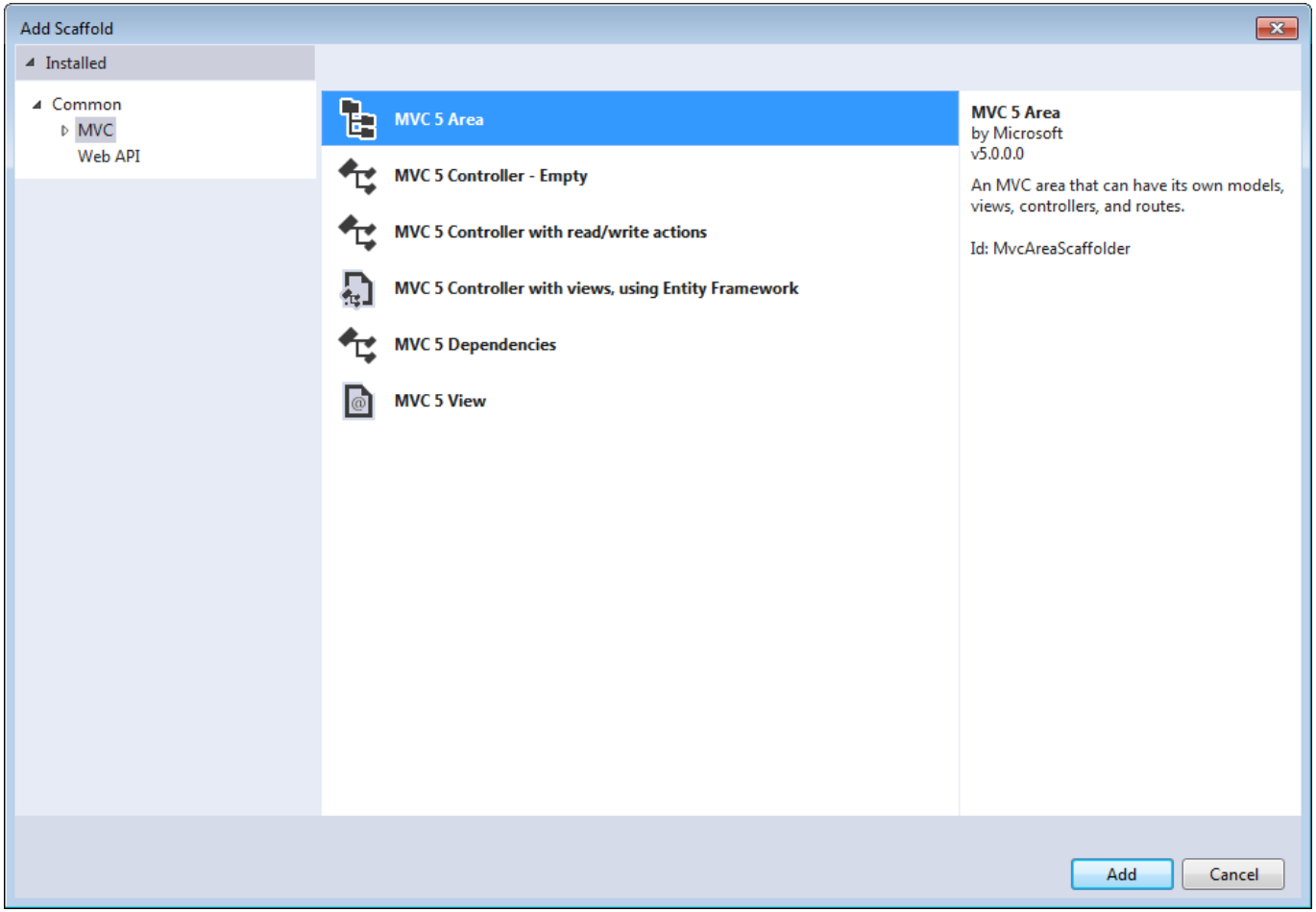
```
...
@foreach (var item in Model) {
<tr>
    <td>
        @Html.DisplayFor(model => item.ID)
    </td>
    <td>
        @Html.ActionLink(item.Name, "Details", new { id = item.ID })
    </td>
    <td>
        @Html.DisplayFor(model => item.Price)
    </td>
    <td>
        <img src="@Url.Content(item.Image)" height="60" width="80" alt="Image" />
    </td>
</tr>
}
</table>
```

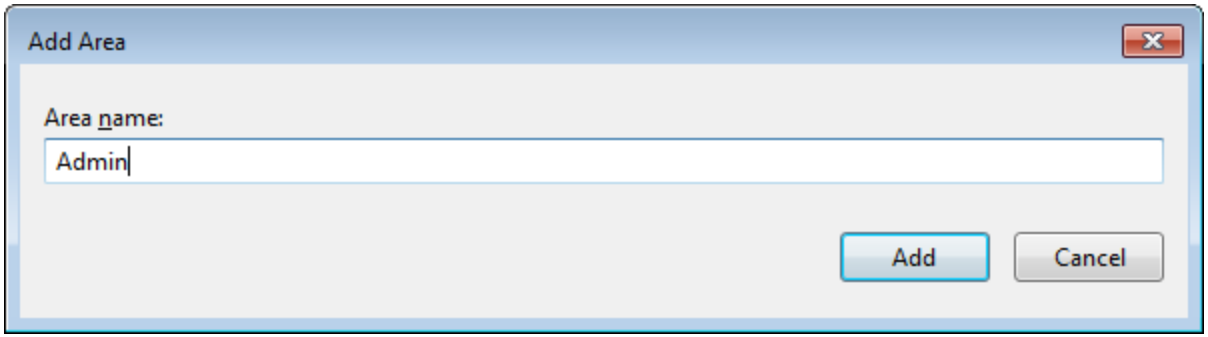
ASP.NET (C#)

Article 09

In this article, I'll explain an easy but an important concept of how to create **Admin** page in ASP.NET.


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'ASP.NET' (1 project)
Article17
Properti
Referenc
App_Dat
App_Sta
Content
Controll
fonts
Models
Scripts
Views
favicon.
Global.a
package
Project_
Web.con
Build
Rebuild
Clean
View
Analyze
Convert
Publish...
Scope to This
New Solution Explorer View
Add
Manage NuGet Packages...
Set as StartUp Project
Debug
Source Control
Cut Ctrl+X
Paste Ctrl+V
Remove Del
Rename
Unload Project
Open Folder in File Explorer
Properties Alt+Enter
Area...
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Scaffolded Item...
New Folder
Add ASP.NET Folder
Reference...
Service Reference...
HTML Page
JavaScript File
Style Sheet
Web Form
MVC 5 View Page (Razor)
Web API Controller Class (v2)
Class...


Add Scaffold
Installed
Common
MVC
Web API
MVC 5 Area
MVC 5 Controller - Empty
MVC 5 Controller with read/write actions
MVC 5 Controller with views, using Entity Framework
MVC 5 Dependencies
MVC 5 View
MVC 5 Area
by Microsoft
v5.0.0.0
An MVC area that can have its own models, views, controllers, and routes.
Id: MvcAreaScaffolder
Add
Cancel


Add Area
Area name:
Admin
Add
Cancel

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'ASP.NET' (1 project)
  - Article17
    - Properties
    - References
    - App_Data
    - App_Start
    - Areas
      - Admin
        - Controllers
        - Models
        - Views
        - AdminAreaRegistration.cs
    - Content
    - Controllers
    - fonts
    - Models
    - Scripts
    - Views
    - favicon.ico
    - Global.asax
    - packages.config
    - Project_Readme.html
    - Web.config

Solution Explorer

Search Solution Explorer (Ctrl+;)

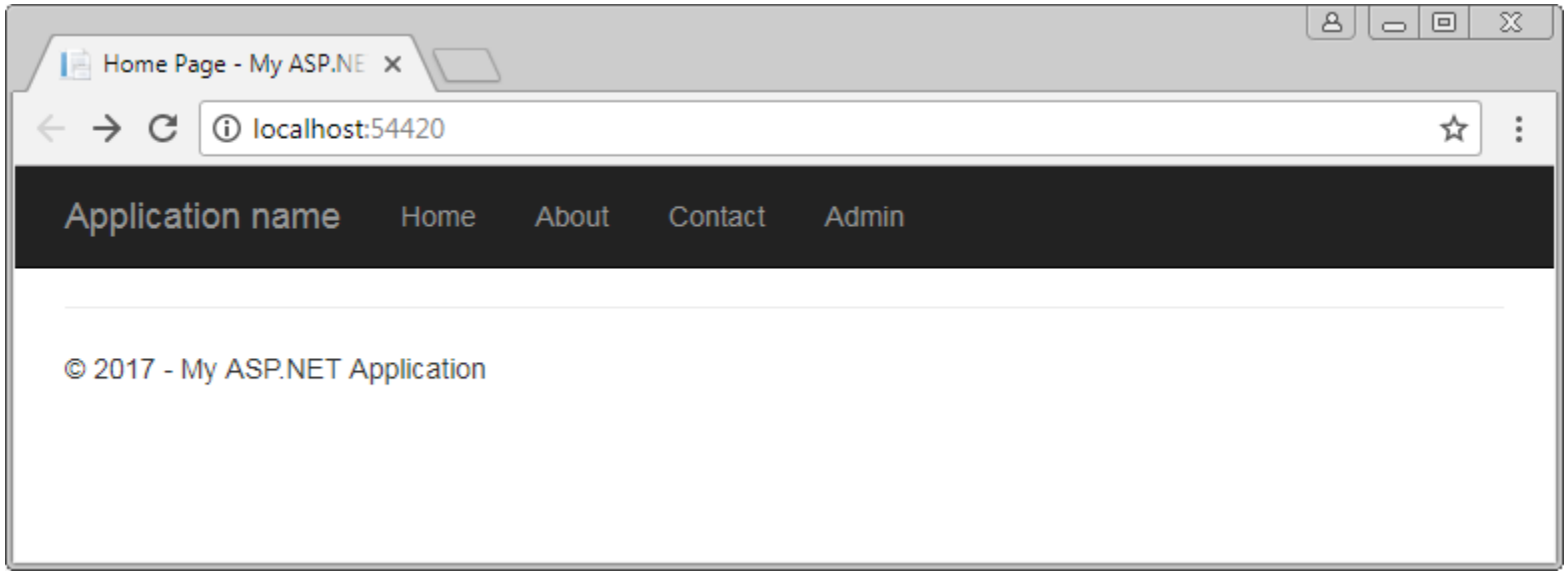
- Solution 'ASP.NET' (1 project)
  - Article17
    - Properties
    - References
    - App_Data
    - App_Start
    - Areas
      - Admin
        - Controllers
          - AdminController.cs
        - Models
        - Views
          - Admin
            - Index.cshtml
          - Shared
            - _Layout.cshtml
            - _ViewStart.cshtml
            - web.config
        - AdminAreaRegistration.cs
    - Content
    - Controllers
    - fonts
    - Models
    - Scripts
    - Views
      - Home
      - Shared
        - _Layout.cshtml
        - Error.cshtml
        - _ViewStart.cshtml
        - Web.config
    - favicon.ico
    - Global.asax
    - packages.config
    - Project_Readme.html
    - Web.config

Solution Explorer | Team Explorer

_Layout.cshtml

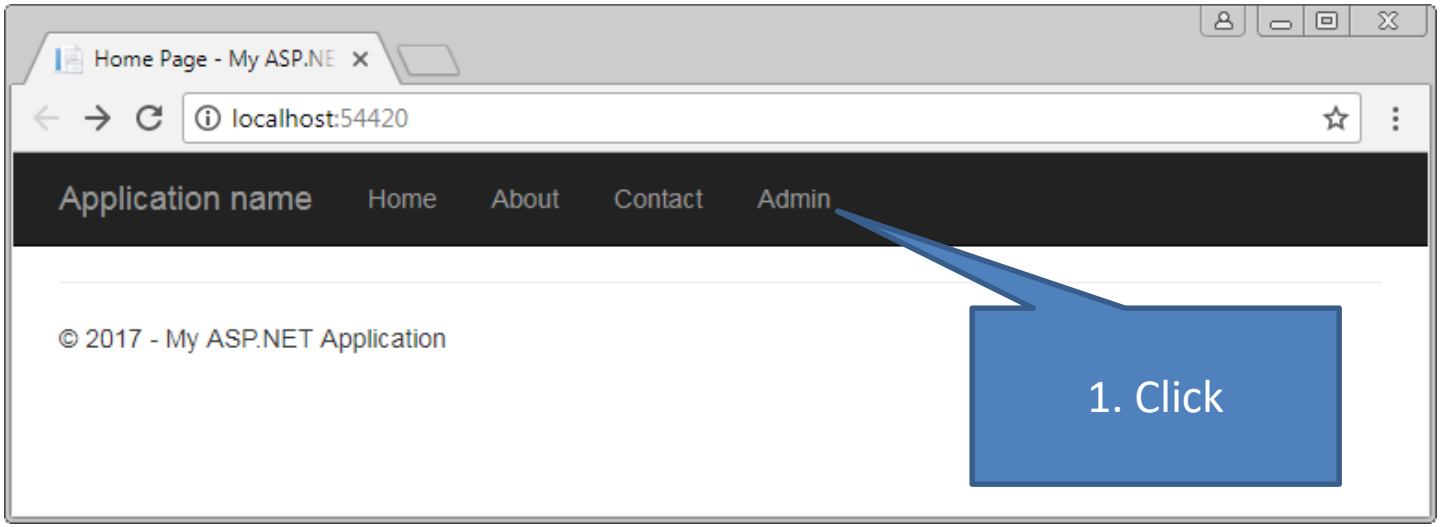
```
. . .
<div class="navbar-collapse collapse">
          <ul class="nav navbar-nav">
          <li>@Html.ActionLink("Home", "Index", "Home")
          <li>@Html.ActionLink("About", "About", "Home")</li>
          <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
          <li>@Html.ActionLink("Admin", "Index", "Admin", new { area = "Admin" }, new { })</li>
          </ul>
</div>
. . .
```

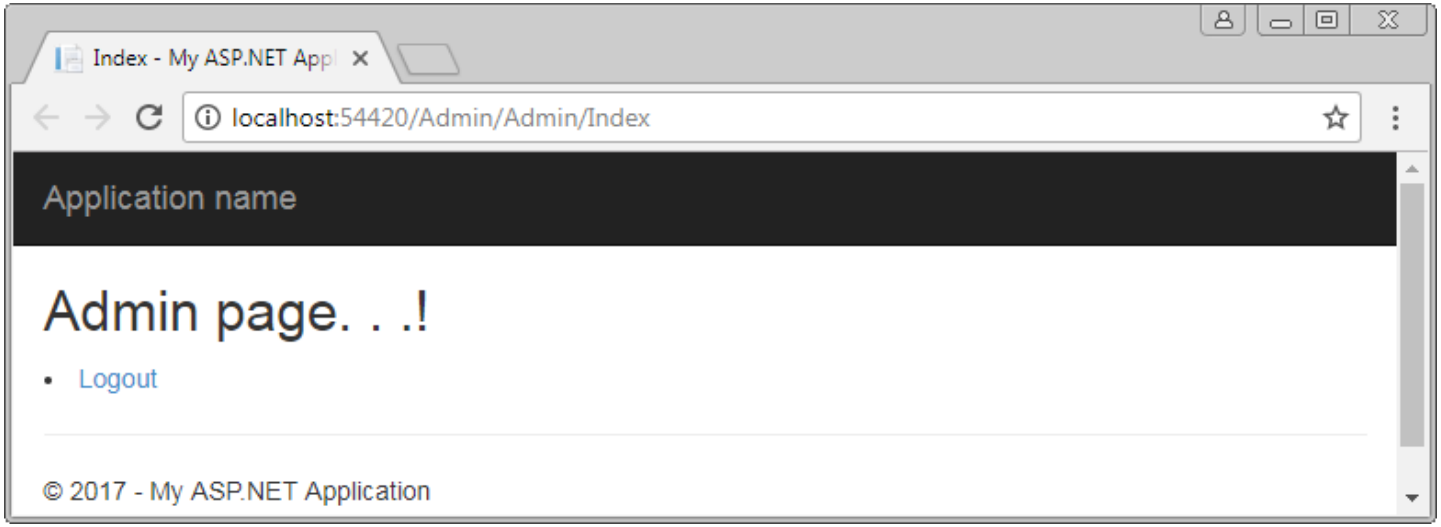
1. Add new ActionLink



Admin\index.cshtml

```
@{
    ViewBag.Title = "Index";
}

<h2>Admin page. . .!</h2>
<li>@Html.ActionLink("Logout", "Index", "Home", new { area = "" }, new { })</li>
```

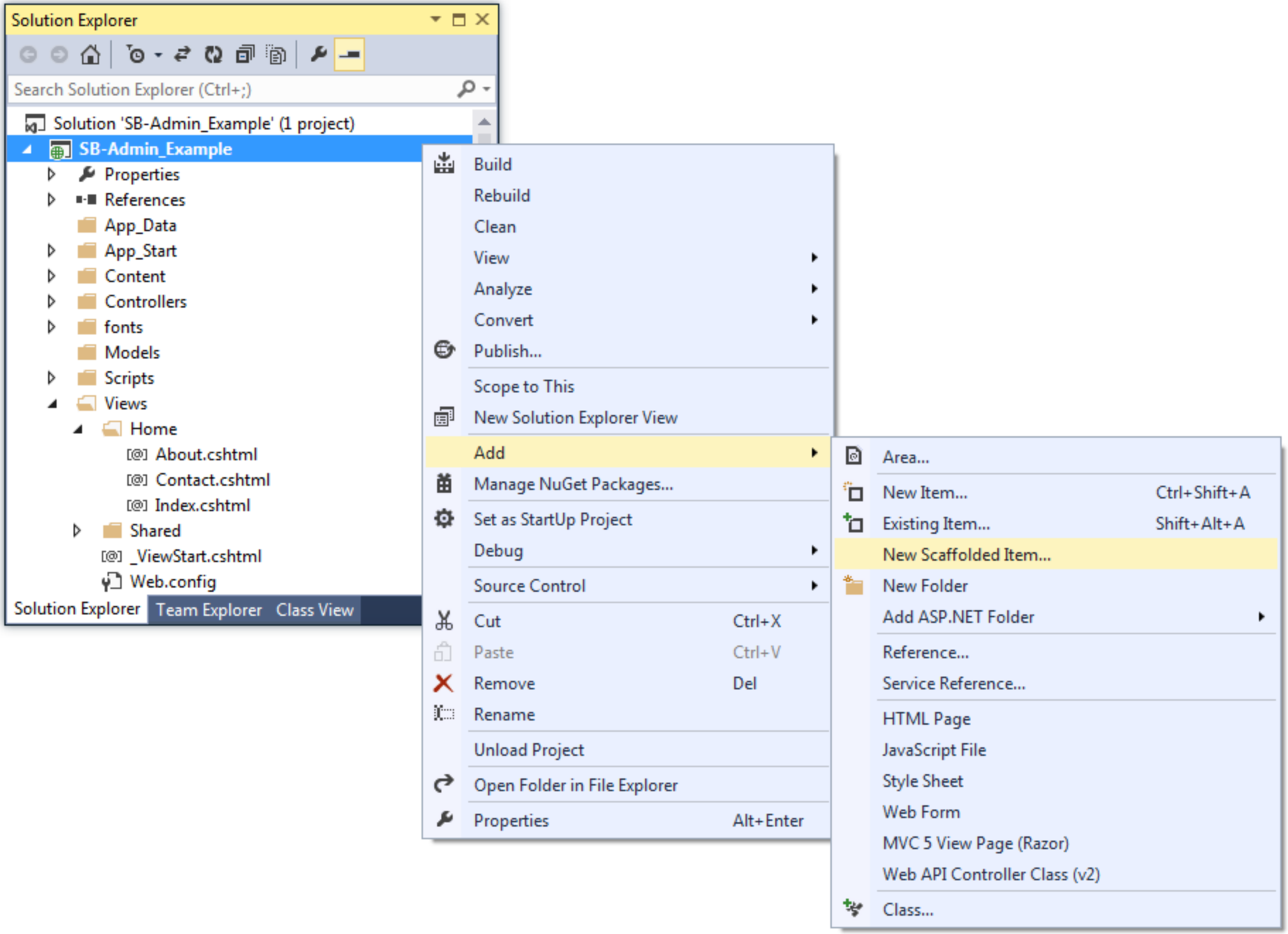


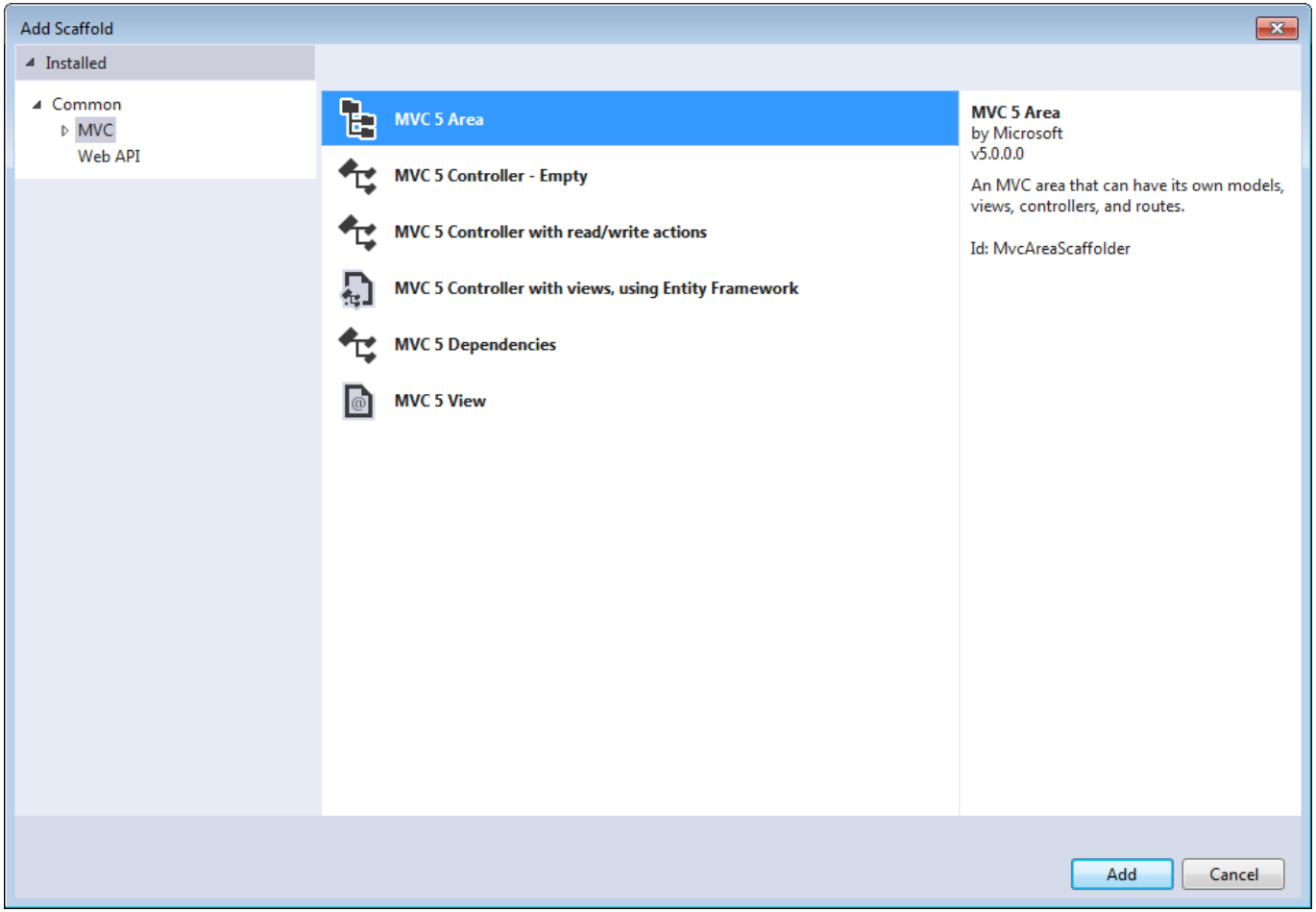


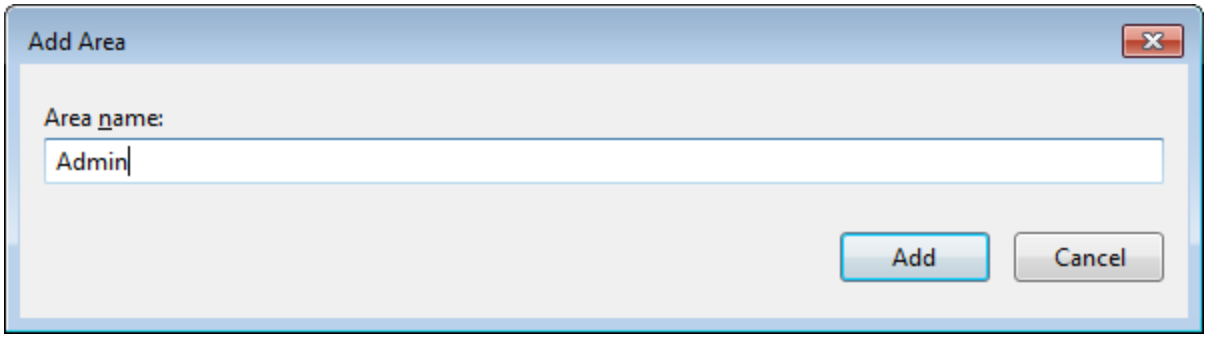
ASP.NET (C#)

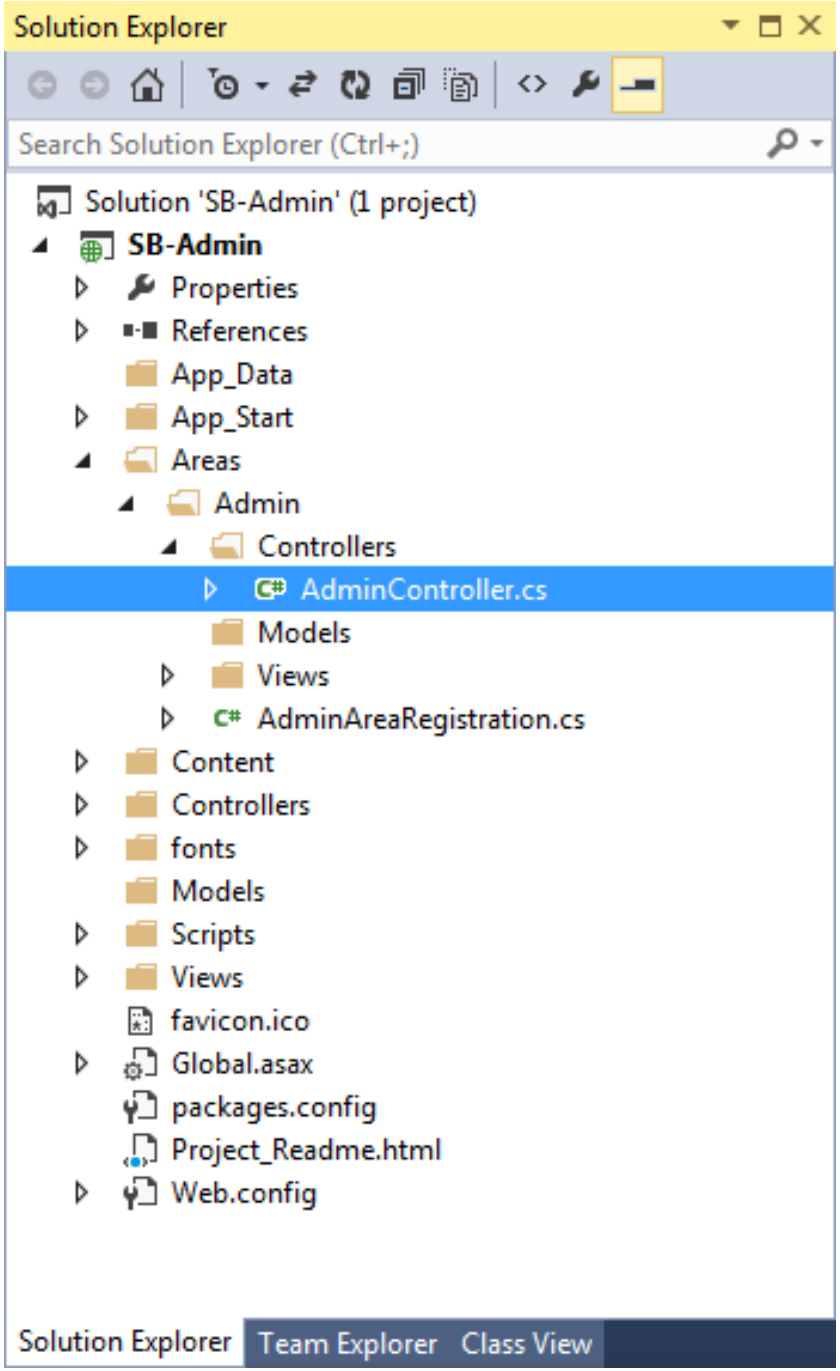
Article 10

In this article, I'll explain an easy but an important concept of how to create **SB-Admin** page in ASP.NET.

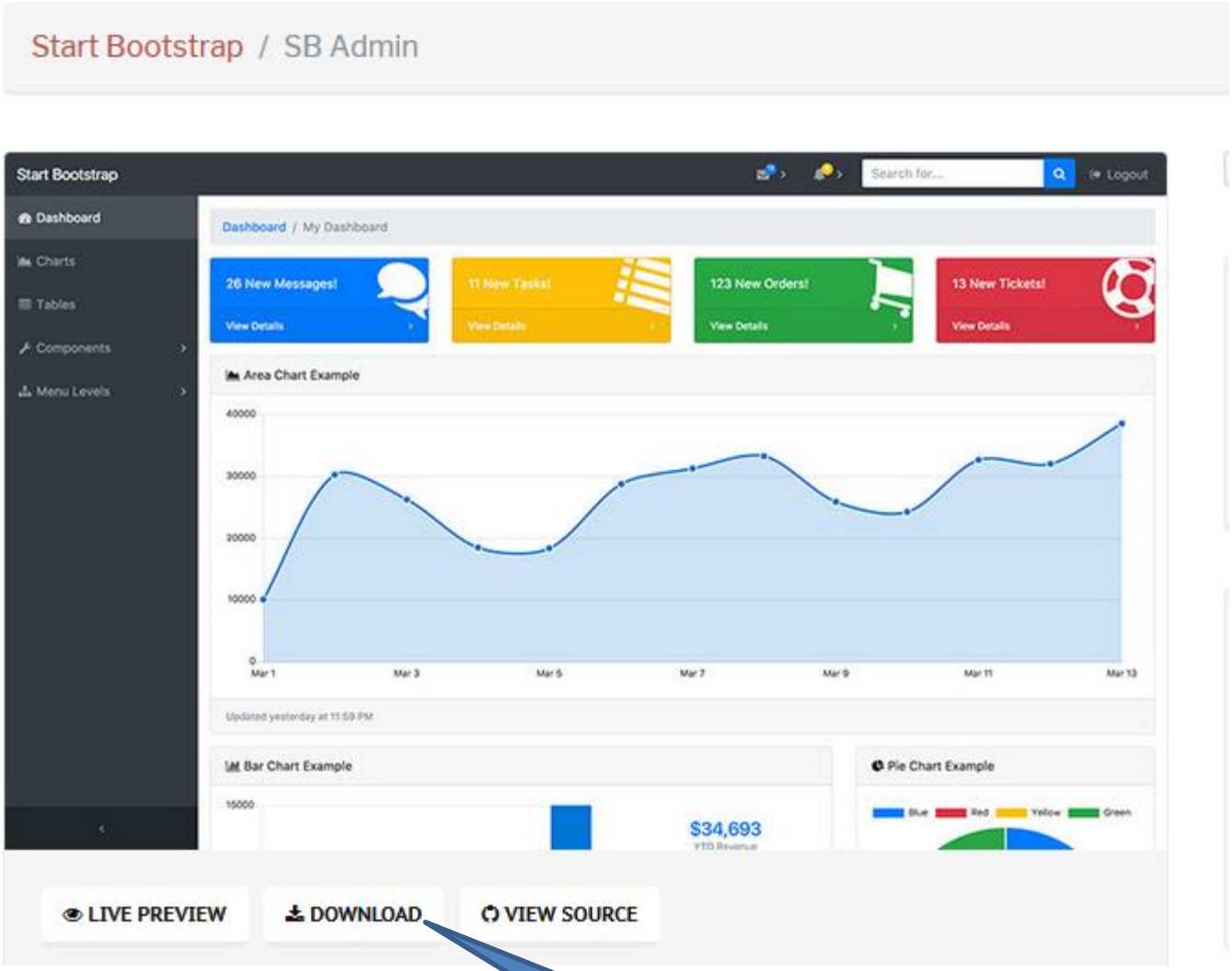

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SB-Admin_Example' (1 project)
SB-Admin_Example
Properties
References
App_Data
App_Start
Content
Controllers
fonts
Models
Scripts
Views
Home
About.cshtml
Contact.cshtml
Index.cshtml
Shared
_ViewStart.cshtml
Web.config
Solution Explorer
Team Explorer
Class View
Build
Rebuild
Clean
View
Analyze
Convert
Publish...
Scope to This
New Solution Explorer View
Add
Manage NuGet Packages...
Set as StartUp Project
Debug
Source Control
Cut Ctrl+X
Paste Ctrl+V
Remove Del
Rename
Unload Project
Open Folder in File Explorer
Properties Alt+Enter
Area...
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Scaffolded Item...
New Folder
Add ASP.NET Folder
Reference...
Service Reference...
HTML Page
JavaScript File
Style Sheet
Web Form
MVC 5 View Page (Razor)
Web API Controller Class (v2)
Class...


Add Scaffold
Installed
Common
MVC
Web API
MVC 5 Area
MVC 5 Controller - Empty
MVC 5 Controller with read/write actions
MVC 5 Controller with views, using Entity Framework
MVC 5 Dependencies
MVC 5 View
MVC 5 Area
by Microsoft
v5.0.0.0
An MVC area that can have its own models, views, controllers, and routes.
Id: MvcAreaScaffolder
Add
Cancel


Add Area
Area name:
Admin
Add
Cancel


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'SB-Admin' (1 project)
SB-Admin
Properties
References
App_Data
App_Start
Areas
Admin
Controllers
AdminController.cs
Models
Views
AdminAreaRegistration.cs
Content
Controllers
fonts
Models
Scripts
Views
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config
Solution Explorer
Team Explorer
Class View

https://startbootstrap.com/template-overviews/sb-admin/



1. Click

« Local Disk (D:) ▸ startbootstrap-sb-admin-gh-pages ▸

Search startbootstrap-sb-admin-gh-p...

File Edit View Tools Help

Organize ▾ Include in library ▾ Share with ▾ New folder

- Favorites
  - Desktop
  - Downloads
  - Recent Places
- Libraries
  - Documents
  - Music
  - Pictures
  - Videos
- Computer
  - SYSTEM (C:)
  - Local Disk (D:)
  - SOFTWARE (E:)
  - CD Drive (L:)
- Network

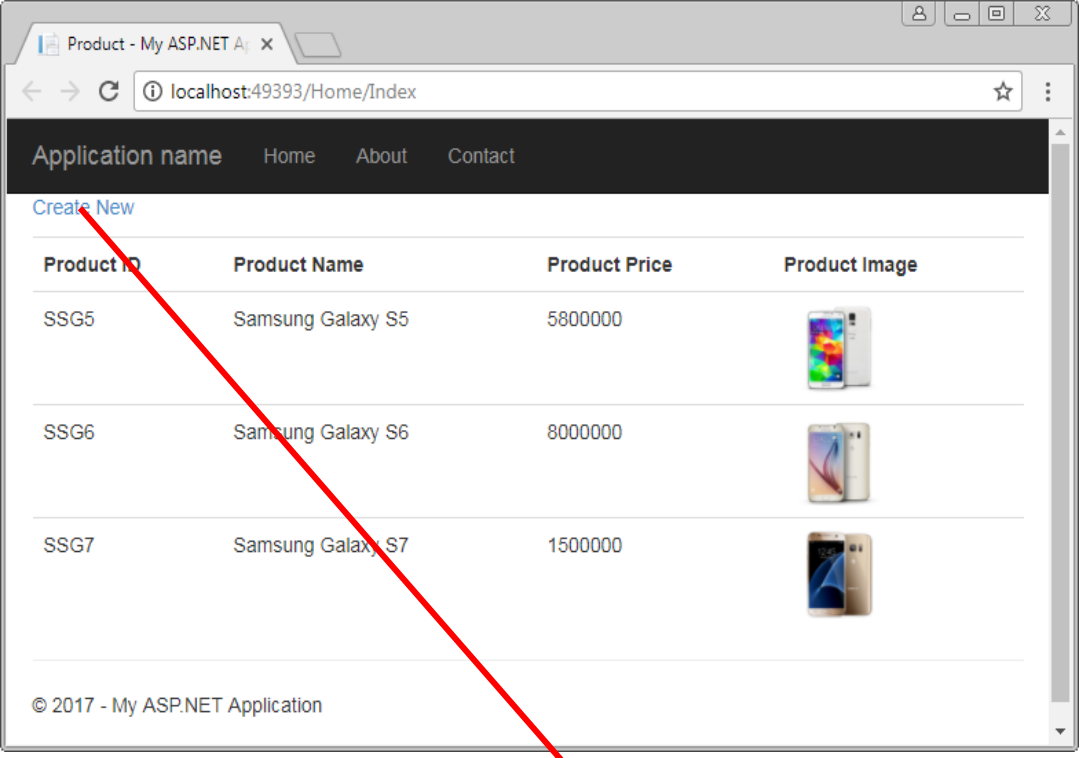
Name	Date modified	Type	Size
css	30/01/2018 10:10 ...	File folder	
js	30/01/2018 10:10 ...	File folder	
pug	30/01/2018 10:10 ...	File folder	
scss	30/01/2018 10:10 ...	File folder	
vendor	30/01/2018 10:10 ...	File folder	
.gitignore	30/01/2018 10:10 ...	GITIGNORE File	1 KB
.jsbeautifyrc	30/01/2018 10:10 ...	JSBEAUTIFYRC File	1 KB
.travis.yml	30/01/2018 10:10 ...	YML File	1 KB
blank	30/01/2018 10:10 ...	Chrome HTML Do...	13 KB
cards	30/01/2018 10:10 ...	Chrome HTML Do...	15 KB
charts	30/01/2018 10:10 ...	Chrome HTML Do...	14 KB
forgot-password	30/01/2018 10:10 ...	Chrome HTML Do...	2 KB
gulpfile	30/01/2018 10:10 ...	JScript Script File	4 KB
index	30/01/2018 10:10 ...	Chrome HTML Do...	45 KB
LICENSE	30/01/2018 10:10 ...	File	2 KB
login	30/01/2018 10:10 ...	Chrome HTML Do...	3 KB
navbar	30/01/2018 10:10 ...	Chrome HTML Do...	14 KB
package.json	30/01/2018 10:10 ...	JSON File	2 KB
package-lock.json	30/01/2018 10:10 ...	JSON File	279 KB
README.md	30/01/2018 10:10 ...	MD File	5 KB
register	30/01/2018 10:10 ...	Chrome HTML Do...	3 KB
tables	30/01/2018 10:10 ...	Chrome HTML Do...	29 KB

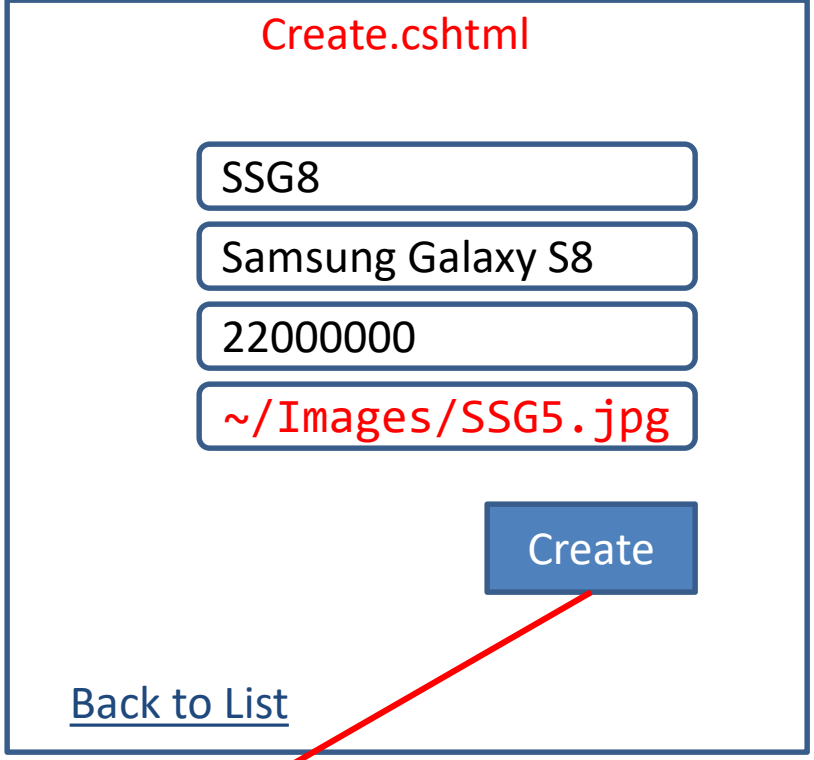
22 items

ASP.NET (C#)

Article 11

In this article, I'll explain an easy but an important concept of how to Add data.






```
public ActionResult Create()
{
        return View();
}
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include="ID, Name,Image, Price")]
Product product)
{
        if (ModelState.IsValid)
```

HomeController

```
private ProductEntity db = new ProductEntity();
public ActionResult Index()
{
        return View(db.Products.ToList());
}
public ActionResult Create()
{
        return View();
}
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include="ID, Name,Image, Price")] Product
product)
{
        if (ModelState.IsValid)
        {
                db.Products.Add(product);
                db.SaveChanges();
                return RedirectToAction("Index");
        }

        return View(product);
}
```

View Index.cshtml

```
@model IEnumerable<Article06.Models.Product>
@{
    ViewBag.Title = "Product";
}
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>Product ID</th>
        <th>Product Name</th>
        <th>Product Price</th>
        <th>Product Image </th>
    </tr>
 . . .
```

View Index.cshtml

```
...
@foreach (var item in Model)
{
        <tr>
            <td>
                @Html.DisplayFor(model => item.ID)
            </td>
            <td>
                @Html.DisplayFor(model => item.Name)
            </td>
            <td>
                @Html.DisplayFor(model => item.Price)
            </td>
            <td>
                <img src="@Url.Content(item.Image)" height="60" width="80"
alt="Image" />
            </td>
         </tr>
}
</table>
```

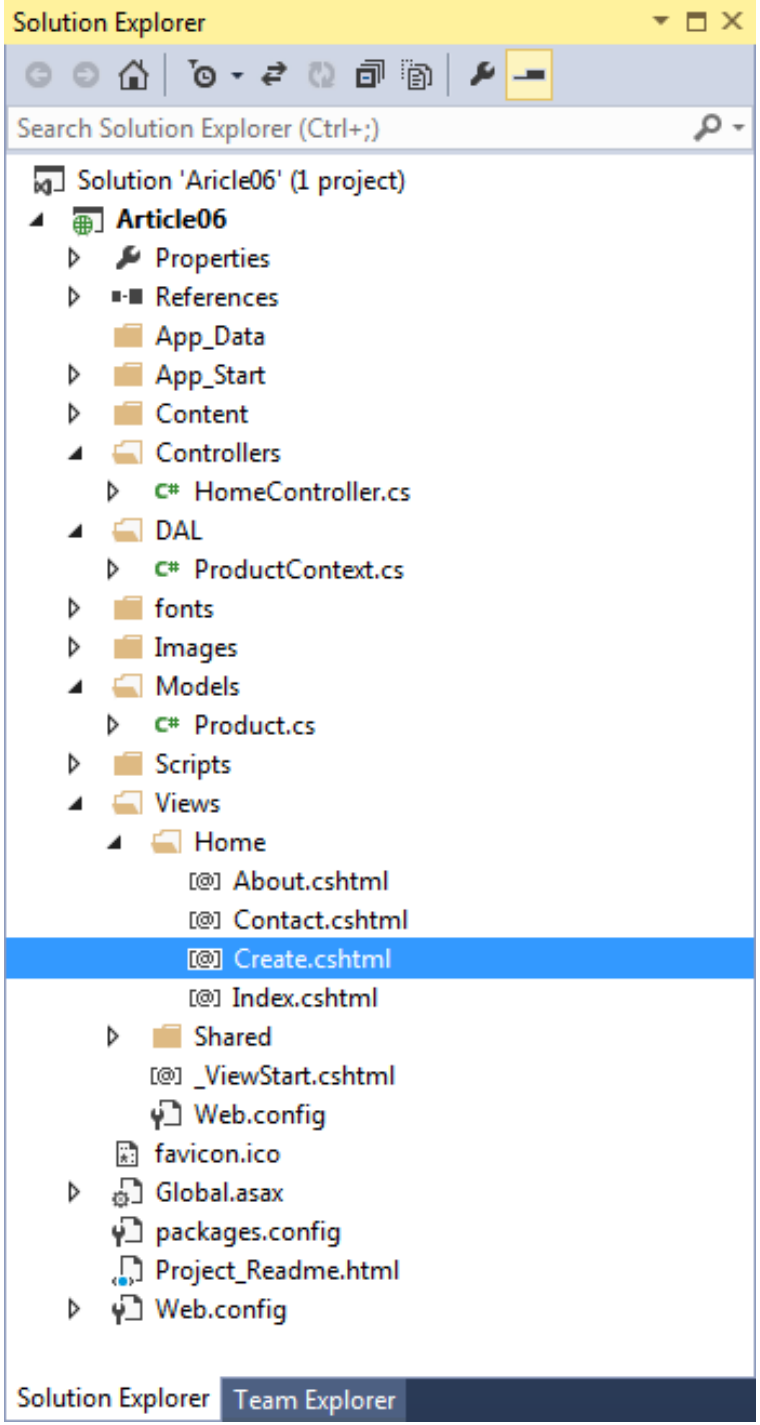
Product - My ASP.NET A

localhost:49393/Home/Index

Application name Home About Contact

Create New

Product ID	Product Name	Product Price	Product Image
SSG5	Samsung Galaxy S5	5800000	
SSG6	Samsung Galaxy S6	8000000	
SSG7	Samsung Galaxy S7	1500000	


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Aricle06' (1 project)
Article06
Properties
References
App_Data
App_Start
Content
Controllers
HomeController.cs
DAL
ProductContext.cs
fonts
Images
Models
Product.cs
Scripts
Views
Home
About.cshtml
Contact.cshtml
Create.cshtml
Index.cshtml
Shared
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config
Solution Explorer
Team Explorer

View Create.cshtml

```
@model Article06.Models.Product
@{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>Product</h4>
        <hr />
        @Html.ValidationSummary(true)
        <div class="form-group">
            @Html.LabelFor(model => model.Image, new { @class = "control-label
col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Image)
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Name, new { @class = "control-label
col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Name)
            </div>
        </div>
```

View Create.cshtml

```
        <div class="form-group">
            @Html.LabelFor(model => model.Price, new { @class = "control-
label col-md-2" })
          <div class="col-md-10">
                @Html.EditorFor(model => model.Price)
          </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default"
/>
            </div>
        </div>
    </div>
}
<div>
    @Html.ActionLink("Back to List", "Index")
</div>
```

ASP.NET (C#)

Article 12

In this article, I'll explain an easy but an important concept of how to Edit data.

View Index.cshtml

```
...
@foreach (var item in Model)
{
        <tr>
            <td>
                @Html.DisplayFor(model => item.ID)
            </td>
            <td>
                @Html.DisplayFor(model => item.Name)
            </td>
            <td>
                @Html.DisplayFor(model => item.Price)
            </td>
            <td>
                <img src="@Url.Content(item.Image)" height="60" width="80"
alt="Image" />
            </td>
            <td>
                 @Html.ActionLink("Edit", "Edit", new { id = item.ID })
            </td>
         </tr>
}
```

Product - My ASP.NET A

localhost:49393

Application name Home About Contact

Create New

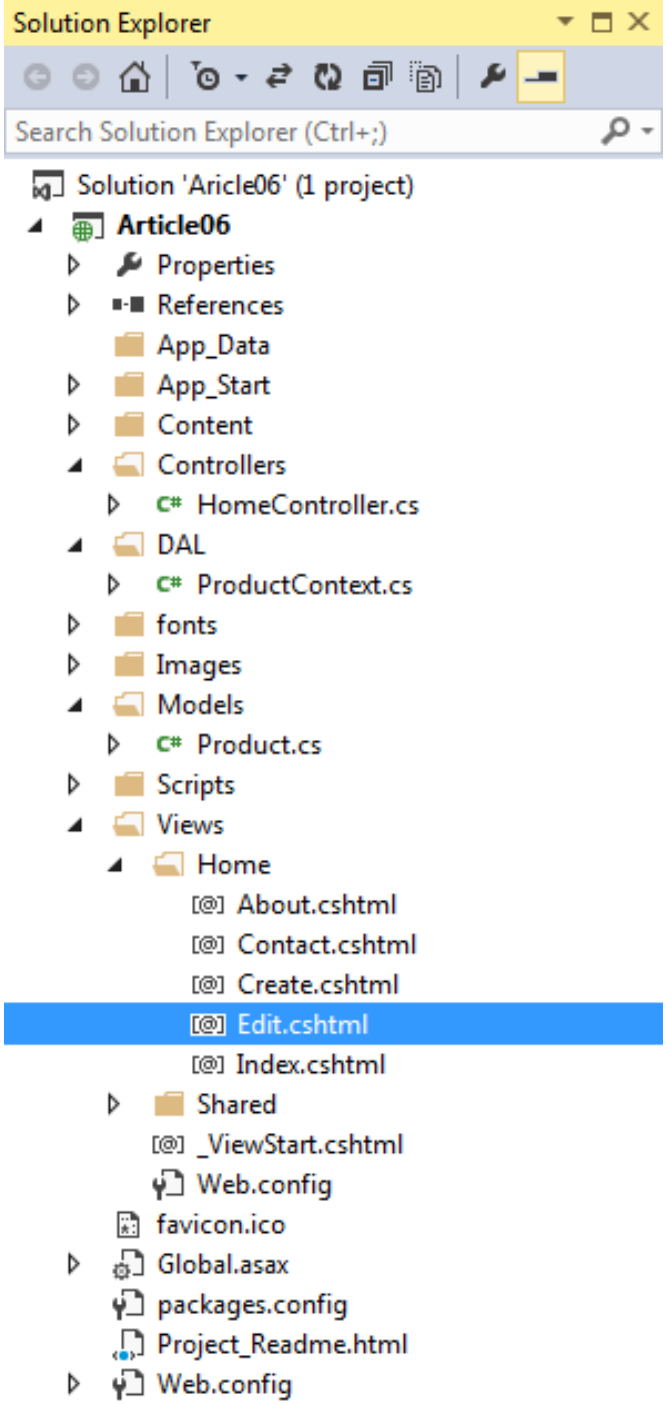
Product ID	Product Name	Product Price	Product Image	
SSG5	Samsung Galaxy S5	5800000		Edit
SSG6	Samsung Galaxy S6	8000000		Edit
SSG7	Samsung Galaxy S7	1500000		Edit

HomeController

```
private ProductEntity db = new ProductEntity();
public ActionResult Index()
{
        return View(db.Products.ToList());
}
. . .
public ActionResult Edit(int? id)
{
        if (id == null)
        {
           return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Product product = db.Products.Find(id);
        if (product == null)
        {
                return HttpNotFound();
        }
        return View(product);
}
```

HomeController

```
private ProductEntity db = new ProductEntity();
public ActionResult Index()
{
        return View(db.Products.ToList());
}
. . .
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include="Id, Name,Image, Price")] Product
product)
{
        if (ModelState.IsValid)
        {
                db.Entry(product).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
        }
        return View(product);
}
```


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Aricle06' (1 project)
Article06
Properties
References
App_Data
App_Start
Content
Controllers
HomeController.cs
DAL
ProductContext.cs
fonts
Images
Models
Product.cs
Scripts
Views
Home
About.cshtml
Contact.cshtml
Create.cshtml
Edit.cshtml
Index.cshtml
Shared
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config

Edit.cshtml

```
@model Article06.Models.Product
@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Edit</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Product</h4>
        <hr />
        @Html.ValidationSummary(true)
        @Html.HiddenFor(model => model.Id)

        <div class="form-group">
            @Html.LabelFor(model => model.Image, new { @class = "control-label
col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Image)
         </div>
        </div>
```

View Edit.cshtml

```
 <div class="form-group">
            @Html.LabelFor(model => model.Name, new { @class = "control-
label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Name)
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.Price, new { @class = "control-
label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Price)
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}
```

ASP.NET (C#)

Article 13

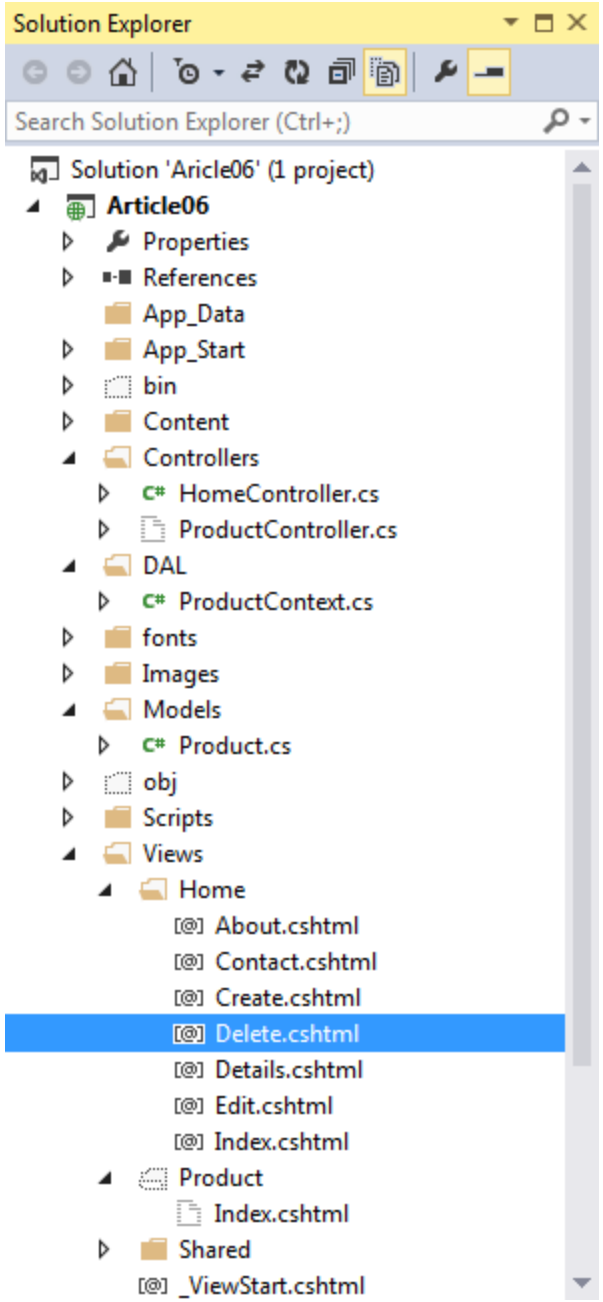
In this article, I'll explain an easy but an important concept of how to Delete data.

View Index.cshtml

```
...
@foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(model => item.ID)
            </td>
            <td>
                @Html.DisplayFor(model => item.Name)
            </td>
            <td>
                @Html.DisplayFor(model => item.Price)
            </td>
            <td>
                <img src="@Url.Content(item.Image)" height="60" width="80"
alt="Image" />
            </td>
            <td>
               @Html.ActionLink("Edit", "Edit", new { id = item.ID }) |
               @Html.ActionLink("Details", "Details", new { id = item.ID }) |
               @Html.ActionLink("Delete", "Delete", new { id = item.ID })
            </td>
       </tr>
}
</table>
```

ProductController

```
private ProductEntity db = new ProductEntity();
public ActionResult Delete(int? id)
{
        if (id == null)
        {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Product product = db.Products.Find(id);
        if (product == null)
        {
                return HttpNotFound();
        }
        return View(product);
}
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
        Product product = db.Products.Find(id);
        db.Products.Remove(product);
        db.SaveChanges();
        return RedirectToAction("Index");
}
```


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Aricle06' (1 project)
Article06
Properties
References
App_Data
App_Start
bin
Content
Controllers
HomeController.cs
ProductController.cs
DAL
ProductContext.cs
fonts
Images
Models
Product.cs
obj
Scripts
Views
Home
About.cshtml
Contact.cshtml
Create.cshtml
Delete.cshtml
Details.cshtml
Edit.cshtml
Index.cshtml
Product
Index.cshtml
Shared
_ViewStart.cshtml

View Delete.cshtml

```
 @model Article06.Models.Product
@{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Product</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Image)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Image)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Name)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Name)
        </dd>
```

View Delete.cshtml

```
 <dt>
            @Html.DisplayNameFor(model => model.Price)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Price)
        </dd>

    </dl>
    @using (Html.BeginForm()) {
        @Html.AntiForgeryToken()
        <div class="form-actions no-color">
            <input type="submit" value="Delete" class="btn btn-default"
/> |
            @Html.ActionLink("Back to List", "Index")
        </div>
    }
</div>
```

ASP.NET (C#)

Article 14

In this article, I'll explain an easy but an important concept of how to One to Many data.

Category.cs

```
public class Category
{
        public string CategoryID { get; set; }
        public string CategoryName { get; set; }

        public virtual ICollection<Product> Products { get; set;
}
```

Product.cs

```
public class Product
{
        public string ID { get; set; }
        public string Name { get; set; }
        public string Image{ get; set; }
        public string Detail{ get; set; }
        public int Price{ get; set; }
        public virtual Category Category { get; set; }
}
```

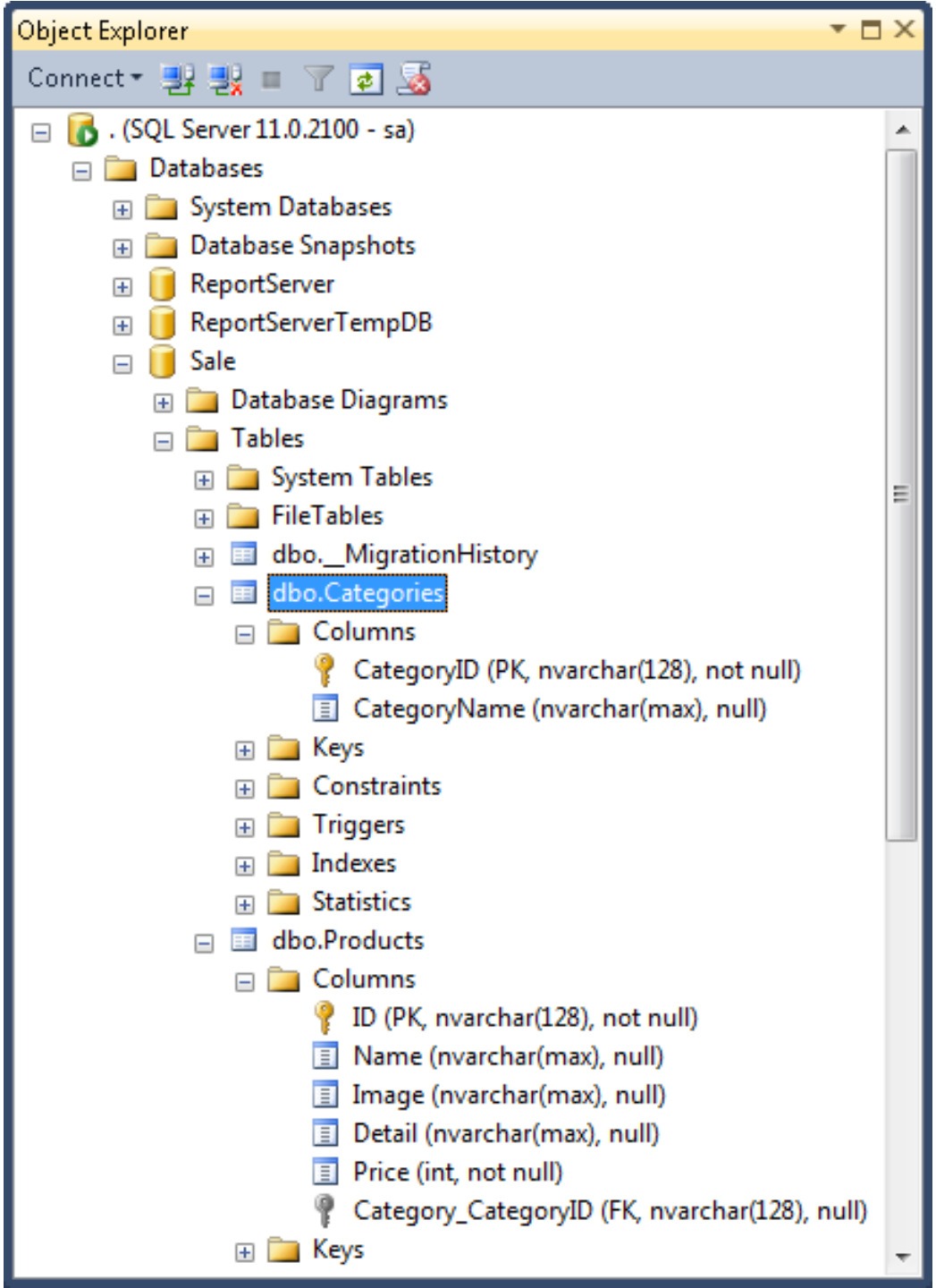
ProductContext.cs

```
namespace Article07.DAL
{
    public class ProductContext : DbContext
    {
        public ProductContext() : base("name=ProductContext")
        {
        }
        public System.Data.Entity.DbSet<Product> Products { get; set; }
        public System.Data.Entity.DbSet<Category> Categories { get; set; }
    }
}
```

HomeController.cs

```
. . .
using Article14.DAL;
public class HomeController : Controller
{
        private ProductContext db = new ProductContext();

        public ActionResult Index()
        {
             return View(db.Categories.ToList());
        }
}
```

F5…Run …


Object Explorer
Connect
. (SQL Server 11.0.2100 - sa)
Databases
System Databases
Database Snapshots
ReportServer
ReportServerTempDB
Sale
Database Diagrams
Tables
System Tables
FileTables
dbo.__MigrationHistory
dbo.Categories
Columns
CategoryID (PK, nvarchar(128), not null)
CategoryName (nvarchar(max), null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Products
Columns
ID (PK, nvarchar(128), not null)
Name (nvarchar(max), null)
Image (nvarchar(max), null)
Detail (nvarchar(max), null)
Price (int, not null)
Category_CategoryID (FK, nvarchar(128), null)
Keys

```sql
use Sale
Go
insert into Categories values ('SS',N'Samsung')
insert into Categories values ('AP',N'Apple')
insert into Categories values ('NO',N'Nokia')
```

```sql
insert into Products values ('SSG5',N'Samsung Galaxy S5','~/Images/SSG5.jpg','Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 1.9 GHz Cortex-A15',5800000, 'SS')

insert into Products values ('SSG6',N'Samsung Galaxy S6','~/Images/SSG6.jpg ','Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.5 GHz Cortex-A16', 8000000, 'SS')

insert into Products values ('NOx81',N'Nokia 8.1 Plus','~/Images/NOx81.jpg ','Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17', 7500000, 'NO')

insert into Products values ('NOx5',N'Nokia 5.1','~/Images/NOx5.jpg ','Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17', 5500000, 'NO')

insert into Products values ('IP7',N'IPhone 7','~/Images/IP7.jpg ',N'Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17', 18500000, 'AP')
```

SQLQuery3.sql - (local).Sale (sa (51))*

```sql
insert into Categories values ('NO',N'Nokia')

insert into Products values ('SSG5',N'Samsung Galaxy S5','~/Images/SSG5.jpg',N'Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 1.9 GHz Cortex-A15',5800000, 'SS')
insert into Products values ('SSG6',N'Samsung Galaxy S6','~/Images/SSG6.jpg ',N'Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.5 GHz Cortex-A16', 8000000, 'SS')
insert into Products values ('NOx81',N'Nokia 8.1 Plus','~/Images/NOx81.jpg ',N'Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17', 7500000, 'NO')
insert into Products values ('NOx5',N'Nokia 5.1','~/Images/NOx5.jpg ',N'Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17', 5500000, 'NO')
insert into Products values ('IP7',N'IPhone 7','~/Images/IP7.jpg ',N'Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17', 18500000, 'AP')

select * from Categories
select * from Products
```

100 %

Results | Messages

	CategoryID	CategoryName
1	AP	Apple
2	NO	Nokia
3	SS	Samsung

	ID	Name	Image	Detail	Price	Category_CategoryID
1	IP7	IPhone 7	~/Images/IP7.jpg	Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17	18500000	AP
2	NOx5	Nokia 5.1	~/Images/NOx5.jpg	Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17	5500000	NO
3	NOx81	Nokia 8.1 Plus	~/Images/NOx81.jpg	Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.9 GHz Cortex-A17	7500000	NO
4	SSG5	Samsung Galaxy S5	~/Images/SSG5.jpg	Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 1.9 GHz Cortex-A15	5800000	SS
5	SSG6	Samsung Galaxy S6	~/Images/SSG6.jpg	Độ phân giải màn hình:1080x1920 pixels, Tốc độ CPU:Quad-core 2.5 GHz Cortex-A16	8000000	SS

Query executed successfully. | (local) (11.0 RTM) | sa (51) | Sale | 00:00:00 | 8 rows

Index.cshtml

```
@model IEnumerable<Article14.Models.Category >
@{
    ViewBag.Title = "Category";
}
<table class="table">
    <tr>
        <th>Product ID</th>
        <th>Product Name</th>
    </tr>
    @foreach (var item in Model){
    <tr>
        <td>
            @Html.DisplayFor(model => item.CategoryID)
        </td>
        <td>
            @Html.DisplayFor(model => item.CategoryName)
        </td>
    </tr>
    }
</table>
```

Application name Home About Contact Admin

Category ID	Category Name
AP	Apple
NO	Nokia
SS	Samsung

```
@model IEnumerable<Article14.Models.Category >
@{
    ViewBag.Title = "Category";
}
<table class="table">
    @foreach (var item in Model){
    <tr>
        <td>
            @Html.DisplayFor(model => item.CategoryName)
        </td>
    </tr>
    <tr>
        @Html.Action("Category", "Home", item)
    </tr>
    }
</table>
```

Index.cshtml

```
public class HomeController : Controller {
        private ProductContext db = new ProductContext();

        public ActionResult Index()
        {
            return View(db.Categories.ToList());
        }

        public ActionResult Category(Category category)
        {
            return View(category.Products);
        }
```

Category.cshtml

```
@model IEnumerable<Article14.Models.Product>
@{
    ViewBag.Title = "Category";
}
<table class="table">
    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(model => item.ID)
            </td>
            <td>
                @Html.DisplayFor(model => item.Name)
            </td>
        </tr>
    }
</table>
```

Application name Home About Contact Admin

Apple

IP7	IPhone 7

© 2019 - My ASP.NET Application

Nokia

NOx5	Nokia 5.1
NOx81	Nokia 8.1 Plus

© 2019 - My ASP.NET Application

Samsung

SSG5	Samsung Galaxy S5
SSG6	Samsung Galaxy S6

© 2019 - My ASP.NET Application

© 2019 - My ASP.NET Application

- Clear all in _ViewStart.cshtml file
- append content to index.cshtml file

```
@{
        Layout = "~/Views/Shared/_Layout.cshtml";
}
```

Index.cshtml

```
@model IEnumerable<Article14.Models.Category >
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
    ViewBag.Title = "Category";
}
<table class="table">
    @foreach (var item in Model){
    <tr>
        <td>
            @Html.DisplayFor(model => item.CategoryName)
        </td>
    </tr>
    <tr>
        @Html.Partial("Category", item.Products)
        or
        @Html.Action("Category", "Home", item);
    </tr>
    }
</table>
```

@Html.Partial: Insert sub view with data

@Html.Action: Insert sub view with data from action

Application name Home About Contact Admin

Apple

IP7	IPhone 7

Nokia

NOx5	Nokia 5.1
NOx81	Nokia 8.1 Plus

Samsung

SSG5	Samsung Galaxy S5
SSG6	Samsung Galaxy S6

ASP.NET (C#)

Article 15

In this article, I'll explain an easy but an important concept of how to Sort data.

```
public ActionResult Index(string sortOrder)
{
        ViewBag.NameSortParm = String.IsNullOrEmpty(sortOrder) ? "info_desc" : "";
        ViewBag.PriceSortParm = sortOrder == "Price" ? "price_desc" : "Price";
        var products = from s in db.Products select s;
        switch (sortOrder)
        {
            case "info_desc":
                products = products.OrderByDescending(s => s.Name);
                break;
            case "Price":
                products = products.OrderBy(s => s.Price);
                break;
            case "price_desc":
                products = products.OrderByDescending(s => s.Price);
                break;
            default:
                products = products.OrderBy(s => s.Name);
                break;
        }
        return View(products.ToList());
    }
```

```
public ActionResult Index(string sortOrder){
        ViewBag.NameSortParm = String.IsNullOrEmpty(sortOrder) ? "info_desc" : "";
        ViewBag.PriceSortParm = sortOrder == "Price" ? "price_desc" : "Price";
        string sql = "select * from Products";
        switch (sortOrder)
        {
            case "info_desc":
                sql += " order by Name DESC";
                break;
            case "Price":
                sql += " order by Price";
                break;
            case "price_desc":
                sql += " order by Price DESC";
                break;
            default:
                sql += " order by Name";
                break;
        }
        List<Product> products = db.Products.SqlQuery(sql).ToList<Product>();
        return View(products.ToList());
}
```

View Index.cshtml

```
@model IEnumerable<Article06.Models.Product>
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Image)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Name)
            @Html.ActionLink("Infomation", "Index", new { sortOrder =
                                                    ViewBag.InfomationSortParm })
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Price)
            @Html.ActionLink("Price", "Index", new { sortOrder =
                                                        ViewBag.PriceSortParm })
        </th>
        <th></th>
    </tr>
```

ASP.NET (C#)

Article 15

In this article, I'll explain an easy but an important concept of how to Search data.

HomeController.cs

```
public ActionResult Index(string sortOrder, string searchString)
{
   ViewBag.NameSortParm = String.IsNullOrEmpty(sortOrder) ? "info_desc" : "";
   ViewBag.PriceSortParm = sortOrder == "Price" ? "price_desc" : "Price";
   var products = from s in db.Products select s;
   if (!String.IsNullOrEmpty(searchString))
   {
      products = products.Where(s => s.Name.Contains(searchString) ||
                                                    s.Price.ToString().Contains(searchString));
   }
   switch (sortOrder)
   {
            case "info_desc":
   . . .


return View(products.ToList());
}
```

HomeController.cs

```
public ViewResult Index(string sortOrder, string searchString)
{
    ViewBag.NameSortParm = String.IsNullOrEmpty(sortOrder) ? "info_desc" : "";
    ViewBag.PriceSortParm = sortOrder == "Price" ? "price_desc" : "Price";
    string sql = "select * from Products";
    if (!String.IsNullOrEmpty(searchString))
    {
         sql += " where Name like '" + searchString + "%' or
                 Price like " + searchString + "%";
    }
    switch (sortOrder)
    {
```

Index.cshtml

```
<p>
    @Html.ActionLink("Create New", "Create")
</p>
@using (Html.BeginForm())
{
    <p>
        Find by Name or Price
@Html.TextBox("SearchString")
        <input type="submit" value="Search" />
    </p>
}
<table class="table">
<tr>
   <th>
```

ASP.NET (C#)

Article 16

In this article, I'll explain an easy but an important concept of how to Paging data.

```
public ViewResult Index(string sortOrder, string currentFilter, string searchString,
int? page){
   ViewBag.NameSortParm = String.IsNullOrEmpty(sortOrder) ? "info_desc" : "";
   ViewBag.PriceSortParm = sortOrder == "Price" ? "price_desc" : "Price";
   var products = from s in db.Products select s;
   if (searchString != null){
         page = 1;
   }
   else{
         searchString = currentFilter;
   }
         ViewBag.CurrentFilter = searchString;
   if (!String.IsNullOrEmpty(searchString)){
      products = products.Where(s => s.Name.Contains(searchString) ||
                           s.Price.ToString().Contains(searchString));
   }
   switch (sortOrder){
   . . .
   int pageSize = 10;
   int pageNumber = (page ?? 1);
   return View(products.ToPagedList(pageNumber, pageSize));
}
```

Index.cshtml

```
@model PagedLipro.IPagedList<Article06.Models.Product>
@using PagedLipro.Mvc;
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}
<h2>Index</h2>
@using (Html.BeginForm("Index", "Home", FormMethod.Get))
{
    <p>
        Find by Infomation and Price
@Html.TextBox("SearchString", ViewBag.CurrentFilter as string)
        <input type="submit" value="Search" />
    </p>
}
```

Index.cshtml

```
<table class="table">
   <tr>
      <th>
         <a>Image</a>
      </th>
      <th>
       @Html.ActionLink("Infomation", "Index", new { sortOrder = ViewBag.InfomationSortParm })
         @Html.ActionLink("Infomation", "Index", new { sortOrder = ViewBag.InfomationSortParm,
currentFilter = ViewBag.CurrentFilter })



      </th>
      <th>
       @Html.ActionLink("Price", "Index", new { sortOrder = ViewBag.PriceSortParm })
         @Html.ActionLink("Price ", "Index", new { sortOrder = ViewBag.PriceSortParm, currentFilter =
ViewBag.CurrentFilter })
      </th>
      <th></th>
   </tr>
```

Index.cshtml

```
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Image)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Name)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Price)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
            @Html.ActionLink("Details", "Details", new { id=item.Id }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.Id })
        </td>
    </tr>
}
</table>
@Html.PagedListPager(Model, page => Url.Action("Index",
    new { page, sortOrder = ViewBag.CurrentSort, currentFilter =
ViewBag.CurrentFilter }))
```

ASP.NET (C#)

Article 17

In this article, I'll explain an easy but an important concept of how to use Editor.

Download a ready-to-use CKEditor package

v4.7.3 • 13-09-2017

Basic Package

Minimal toolbar.
For quick input fields.
17 Plugins

Download

Standard Package

Standards-compliant toolbar.
Covers most editing needs
for semantic-driven websites.
48 Plugins

Download

Full Package

Source
Styles
Heading 1

A heavier preset
with a multiline toolbar.
72 Plugins

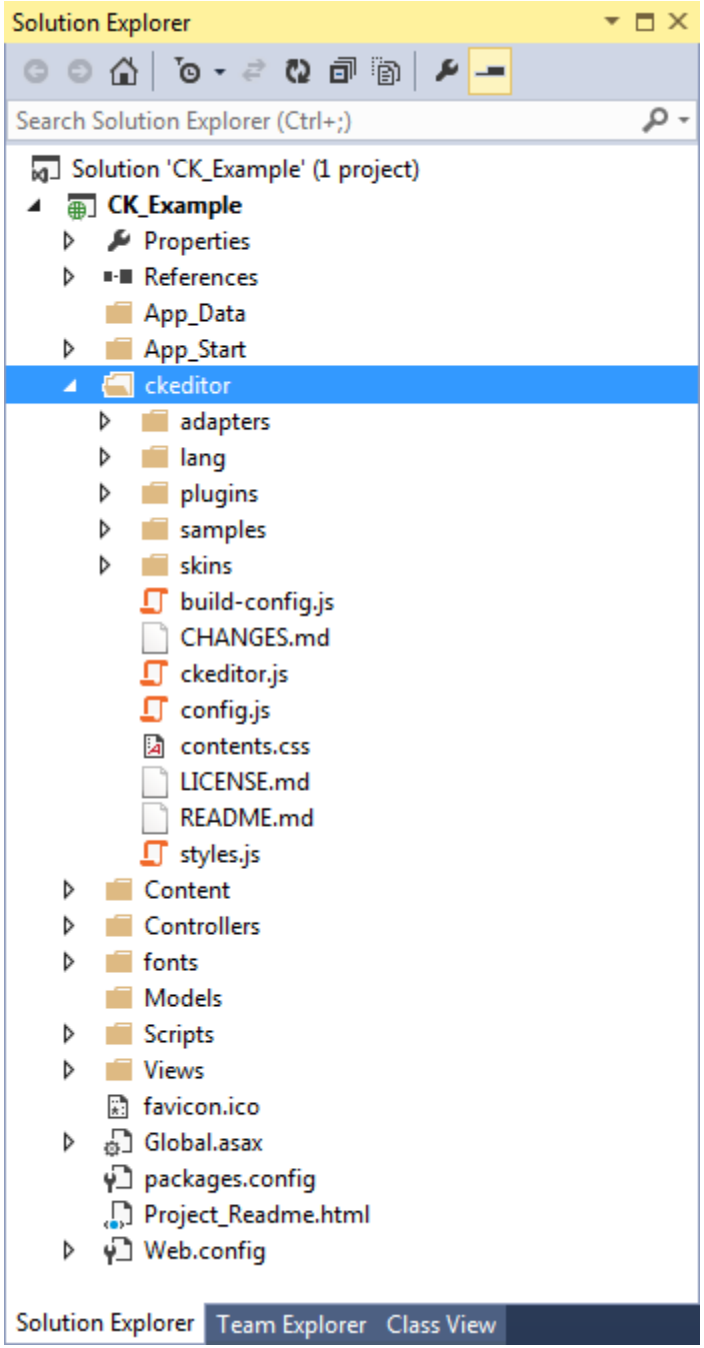
Download

Customize

Heading 1

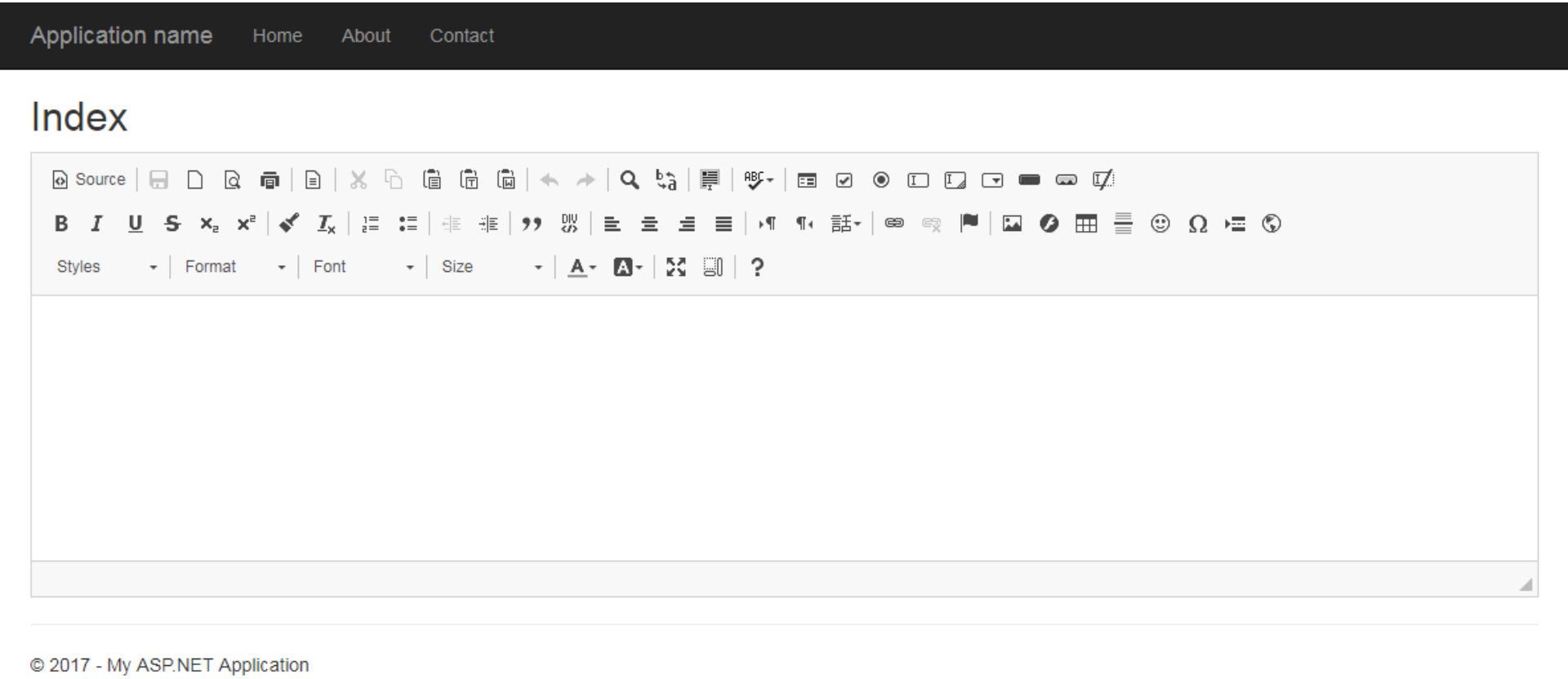
Build your own version
in 3 easy steps.

Online Builder


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'CK_Example' (1 project)
CK_Example
Properties
References
App_Data
App_Start
ckeditor
adapters
lang
plugins
samples
skins
build-config.js
CHANGES.md
ckeditor.js
config.js
contents.css
LICENSE.md
README.md
styles.js
Content
Controllers
fonts
Models
Scripts
Views
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config
Solution Explorer
Team Explorer
Class View

Index.cshtml

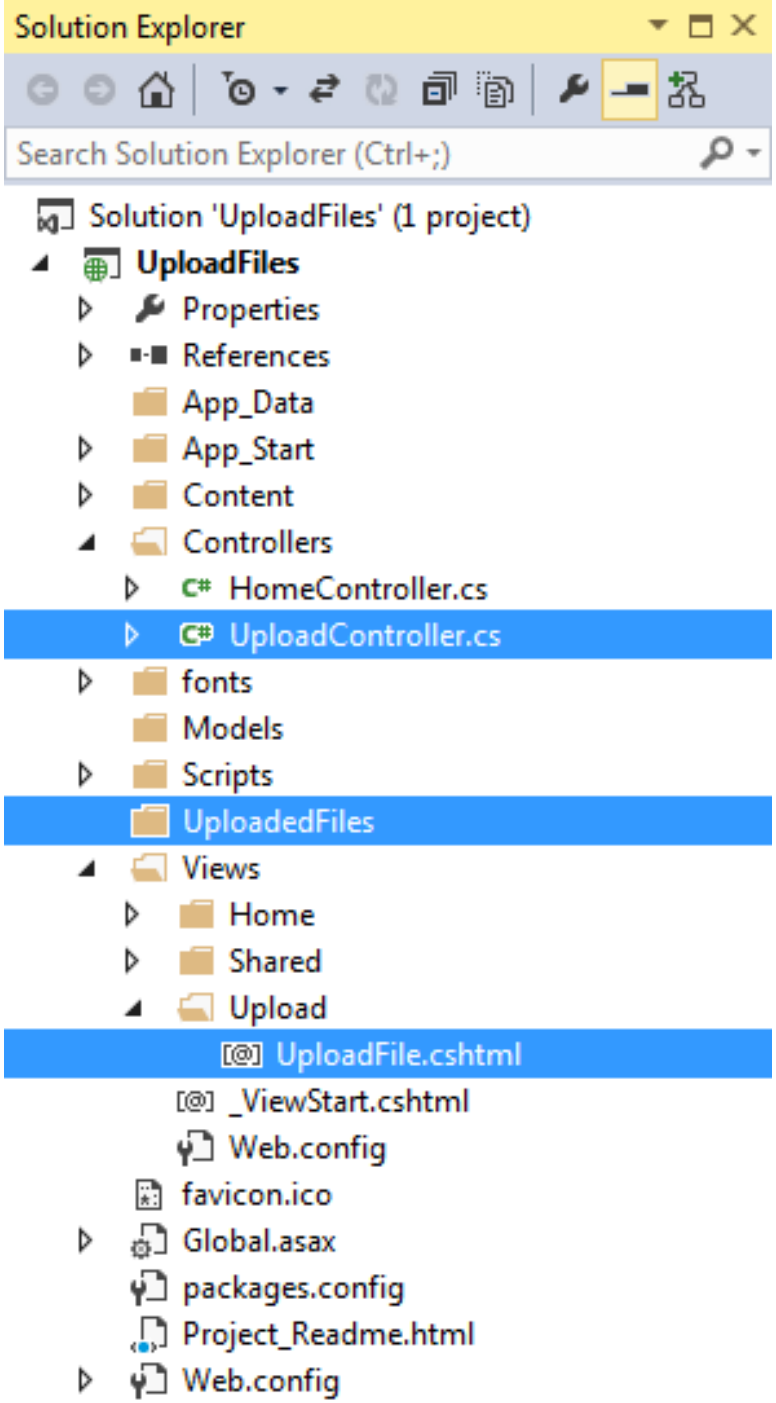
```
@{
    ViewBag.Title = "Home Page";
}
<script src="~/ckeditor/ckeditor.js"></script>
<h2>Index</h2>
<div>@Html.TextArea("editor", new {@class = "ckeditor", @id = "sampleEditor"})</div>
```



ASP.NET (C#)

Article 18

In this article, I'll explain an easy but an important concept of how to Upload File.


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'UploadFiles' (1 project)
UploadFiles
Properties
References
App_Data
App_Start
Content
Controllers
HomeController.cs
UploadController.cs
fonts
Models
Scripts
UploadedFiles
Views
Home
Shared
Upload
UploadFile.cshtml
_ViewStart.cshtml
Web.config
favicon.ico
Global.asax
packages.config
Project_Readme.html
Web.config

```
public class UploadController: Controller  {
[HttpGet]
public ActionResult UploadFile(){
            return View();
}
[HttpPost]
public ActionResult UploadFile(HttpPostedFileBase file){
        try {
                if (file.ContentLength > 0) {
                    string _FileName = Path.GetFileName(file.FileName);
                    string _path =
Path.Combine(Server.MapPath("~/UploadedFiles"), _FileName);
                    file.SaveAs(_path);
                }
                    ViewBag.Message = "File Uploaded Successfully!!";
                return View();
        }
        catch {
                ViewBag.Message = "File upload failed!!";
                return View();
        }
}
```

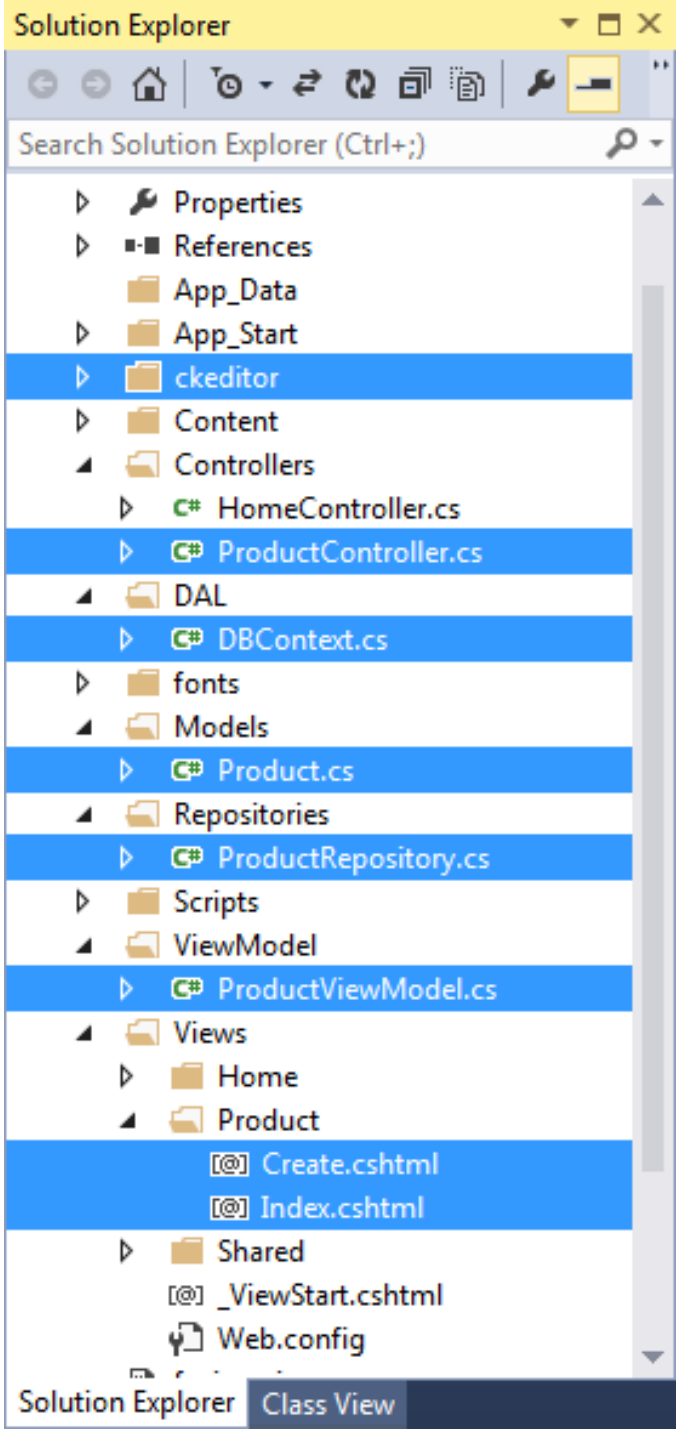
UploadFile.cshtml

```
@{
    ViewBag.Title = "UploadFile";
}

<h2>UploadFile</h2>

@using(Html.BeginForm("UploadFile","Upload", FormMethod.Post,
new { enctype="multipart/form-data"}))
{

    <div>
        @Html.TextBox("file", "", new {  type= "file"}) <br />

        <input type="submit" value="Upload" />

        @ViewBag.Message

    </div>

}
```

ASP.NET (C#)

Article 19

In this article, I'll explain an easy but an important concept of how to Upload File to database.


Solution Explorer
Search Solution Explorer (Ctrl+;)
Properties
References
App_Data
App_Start
ckeditor
Content
Controllers
HomeController.cs
ProductController.cs
DAL
DBContext.cs
fonts
Models
Product.cs
Repositories
ProductRepository.cs
Scripts
ViewModel
ProductViewModel.cs
Views
Home
Product
Create.cshtml
Index.cshtml
Shared
_ViewStart.cshtml
Web.config
Solution Explorer
Class View

ProductModel

```
public class Product
{
        [Key]
        public int ID { get; set; }
        public string Name{ get; set; }
        public int Price{ get; set; }
        public byte[] Image { get; set; }
}
```

ProductViewModel

```
public class ProductViewModel
{
        public int ID { get; set; }
         [Required]
         public string Name{ get; set; }
         [Required]
         public int Price{ get; set; }
         [Required]
         public byte[] Image { get; set; }
}
```

DBContext

```
public class DBContext : DbContext
{
        public DBContext() : base("Name=ProductContext")
        {
            Database.SetInitializer<DBContext>(new
              DropCreateDatabaseIfModelChanges<DBContext>());
        }
        public DbSet<Product> Products { get; set; }
}
```

ProductRepository

```
public class ProductRepository {
        private readonly DBContext db = new DBContext();
        public int UploadImageInDataBase(HttpPostedFileBase file,
                                         ProductViewModel productViewModel)
        {
            productViewModel.Image = ConvertToBytes(file);
            var Product = new Product
            {
                Title = productViewModel.Title,
                Description = productViewModel.Description,
                Contents = productViewModel.Contents,
                Image = productViewModel.Image
            };
            db.Products.Add(Product);
            int i = db.SaveChanges();
            if (i == 1)
            {
                return 1;
            }
            else
            {
                return 0;
            }

        }
```

```
public byte[] ConvertToBytes(HttpPostedFileBase image)
{
        byte[] imageBytes = null;
        BinaryReader reader = new
                    BinaryReader(image.InputStream);
        imageBytes =
              reader.ReadBytes((int)image.ContentLength);
        return imageBytes;
}
```

HomeController

```
public class ProductController : Controller{
        private DBContext db = new DBContext();
        [Route("Index")]
        [HttpGet]
        public ActionResult Index(){
            var Product = db.Products.Select(s => new
            {
                s.ID,
                s.Title,
                s.Image,
                s.Contents,
                s.Description
            });

            List<ProductViewModel> contentModel = Product.Select(item => new
ProductViewModel()
            {
                ID = item.ID,
                Title = item.Title,
                Image = item.Image,
                Description = item.Description,
                Contents = item.Contents
            }).ToList();
            return View(contentModel);
        }
```

```
public ActionResult RetrieveImage(int id){
            byte[] cover = GetImageFromDataBase(id);
            if (cover != null)
            {
                return File(cover, "image/jpg");
            }
            else
            {
                return null;
            }
        }

        public byte[] GetImageFromDataBase(int Id){
            var q = from temp in db.Products where temp.ID == Id select
temp.Image;
            byte[] cover = q.First();
            return cover;
        }

        [HttpGet]
        public ActionResult Create(){
            return View();
        }
}
```

```
[Route("Create")]
[HttpPost]
public ActionResult Create(ProductViewModel model){
        HttpPostedFileBase file = Request.Files["ImageData"];
        ProductRepository service = new ProductRepository ();
        int i = service.UploadImageInDataBase(file, model);
        if (i == 1)
        {
                return RedirectToAction("Index");
        }
        return View(model);
}
```

View/Home/Index

```
@model IEnumerable<Article06.ViewModel.ProductViewModel>
@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New Product", "Create",null, new {
@class="btn btn-primary"})
</p>

<br />
```

```html
<table class="table" style="width: 1200px;">
    <tr>
        <th>
            <b>Name</b>
        </th>
        <th>
            <b>Price</b>
        </th>
        <th>
            <b>Image</b>
        </th>
    </tr>
```

```
@foreach (var item in Model)
{
      <tr>
         <td>
              @Html.DisplayFor(modelItem => item.Name)
         </td>
         <td>
              @Html.Raw(item.Price)
         </td>
         <td>
             <img src="/Product/RetrieveImage/@item.ID" alt=""
height=100 width=200/>
         </td>
      </tr>
  }
</table>
```

View/Create

```
@model Article.ViewModel.ProductViewModel
@{
    ViewBag.Title = "Create";
}
<script src="~/ckeditor/ckeditor.js"></script>
<script src="~/ckeditor/adapters/jquery.js"></script>

<h2>Create New Product With Image</h2>

<script type="text/javascript">
    function fileCheck(obj) {
        var fileExtension = ['jpeg', 'jpg', 'png', 'gif', 'bmp'];
        if ($.inArray($(obj).val().split('.').pop().toLowerCase(),
fileExtension) == -1) {
            alert("Only '.jpeg','.jpg', '.png', '.gif', '.bmp'
formats are allowed.");
        }
    }
</script>
```

```
@using (Html.BeginForm("Create", "Home", FormMethod.Post, new {
enctype ="multipart/form-data" }))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Product</h4>
        <hr />
        @Html.ValidationSummary(true)

        <div class="form-group">
            @Html.LabelFor(model => model.Title, new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.TextBoxFor(model => model.Title, new {
@class = "form-control", placeholder = "Product Title" })
                @Html.ValidationMessageFor(model => model.Title)
            </div>
        </div>
```

```
 <div class="form-group">
            @Html.LabelFor(model => model.Image, new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                <input type="file" name="ImageData" id="ImageData"
onchange="fileCheck(this);" />
                @Html.ValidationMessageFor(model => model.Image)
            </div>
</div>

<div class="form-group">
            @Html.LabelFor(model => model.Description, new { @class
= "control-label col-md-2" })
            <div class="col-md-10">
                @Html.TextAreaFor(model => model.Description, new {
@class = "form-control", placeholder = "Product Description" })
                @Html.ValidationMessageFor(model =>
model.Description)
            </div>
</div>
```

```
 <div class="form-group">
            @Html.LabelFor(model => model.Contents, new { @class
= "control-label col-md-2" })
            <div class="col-md-10">
                @Html.TextAreaFor(model => model.Contents, new {
@class = "ckeditor", placeholder = "Product" })
                @Html.ValidationMessageFor(model =>
model.Contents)
            </div>
</div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value= "Save" class="btn
btn-default" />
            </div>
        </div>
</div>
}
```

Web.config

```
<configuration>

 . . .

 <connectionStrings>
    <add name= "ProductContext"
providerName="System.Data.SqlClient" connectionString="Data
Source=(local);Initial Catalog=Sale;Integrated Security=False;
User Id=sa;Password=sql2012; MultipleActiveResultSets=True" />
  </connectionStrings>
</configuration>
```

ASP.NET (C#)

Article 20

In this article, I'll explain an easy but an important concept of how to use **Authentication** in ASP.NET.

New ASP.NET Project - Article20

Select a template:

Empty | Web Forms | MVC | Web API | Single Page Application | Facebook

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Learn more

Add folders and core references for:

☐ Web Forms ☑ MVC ☐ Web API

☐ Add unit tests

Test project name: Article20.Tests

Change Authentication

Authentication: **Individual User Accounts**

Change Authentication

○ No Authentication

◉ Individual User Accounts

○ Organizational Accounts

○ Windows Authentication

For applications that store user profiles in a SQL Server database. Users can register, or sign in using their existing account for Facebook, Twitter, Google, Microsoft, or another provider.

Learn more

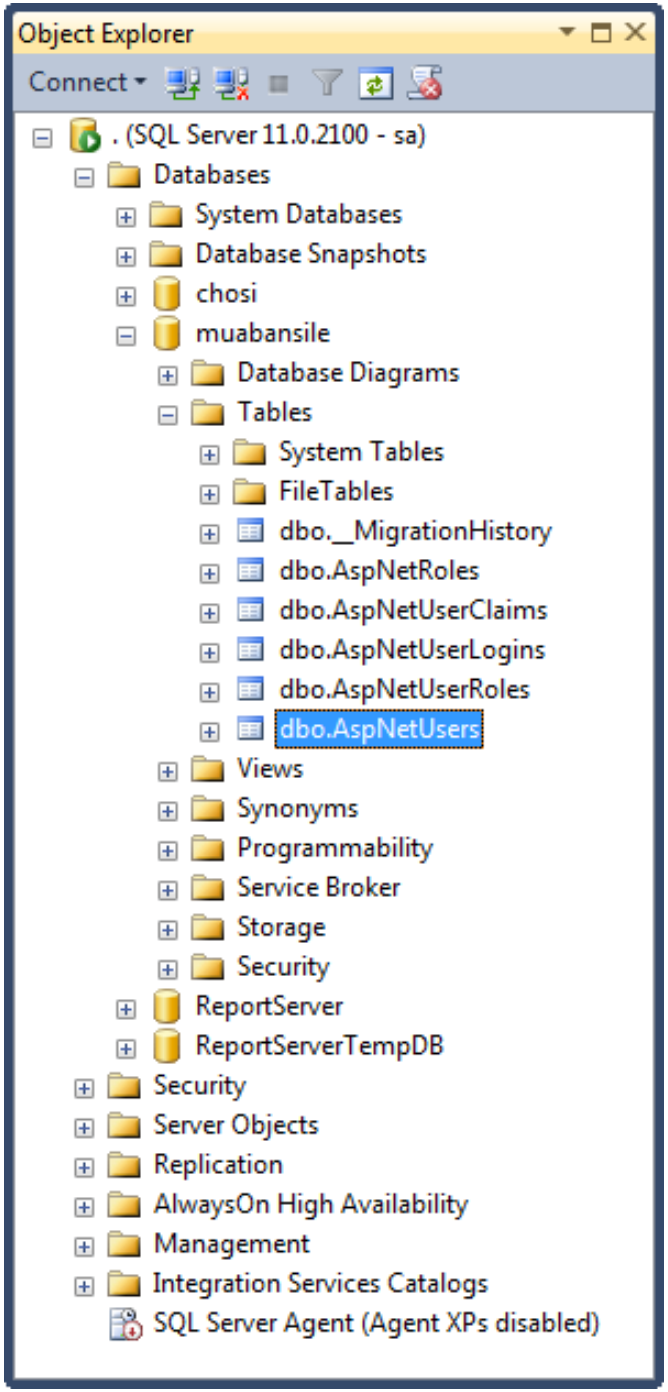
OK | Cancel

```
<connectionStrings>
    <add name="DefaultConnection" connectionString="Data
Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\aspnet-Article20-
20180611012208.mdf;Initial Catalog=aspnet-Article20-20180611012208;Integrated
Security=True"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
```

```
<configuration>

 . . .

 <connectionStrings>
    <add name= "DefaultConnection"
providerName="System.Data.SqlClient" connectionString="Data
Source=(local);Initial Catalog=Sale;Integrated Security=False;
User Id=sa;Password=sql2012; MultipleActiveResultSets=True" />
  </connectionStrings>
</configuration>
```

Run…Test

- Create new user
- Check database



Header View

```
@Html.Partial("_LoginPartial")
```

View

```
@using Microsoft.AspNet.Identity
@if (Request.IsAuthenticated)
{
 . . .
}
else
{
    <ul class="nav navbar-nav navbar-right">
            <li>@Html.ActionLink("Dang ky", "Register", "Account", routeValues:
    null, htmlAttributes: new { id = "registerLink" })</li>
            <li>@Html.ActionLink("Dang nhap", "Login", "Account", routeValues:
    null, htmlAttributes: new { id = "loginLink" })</li>
    </ul>
}
```

ASP.NET (C#)

Article 21

In this article, I'll explain an easy but an important concept of how to use Session and Cookie in ASP.NET.

Session vs Cookie

- Cookies and Sessions are used to store information.

- Cookies are only stored on the client-side machine, while sessions get stored on the client as well as a server.

Session

- A session creates a file in a temporary directory on the server where registered session variables and their values are stored. This data will be available to all pages on the site during that visit.
- A session ends when the user closes the browser or after leaving the site, the server will terminate the session after a predetermined period of time, commonly 30 minutes duration.

How to use Session

- Delare Session

```
public ActionResult Login()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Login(UserProfile objUser)
{
    if (ModelState.IsValid)
    {
        using(DB_Entities db = new DB_Entities())
        {
            var obj = db.UserProfiles.Where(a => a.UserName.Equals(objUs
            if (obj != null)
            {
                Session["UserID"] = obj.UserId.ToString();
                Session["UserName"] = obj.UserName.ToString();
                return RedirectToAction("UserDashBoard");
            }
        }
    }
    return View(objUser);
}
```

How to use Session

- Take Session

```
public ActionResult UserDashBoard()
{
    if (Session["UserID"] != null)
    {
        return View();
    } else
    {
        return RedirectToAction("Login");
    }
}
```

Cookies

- Cookies are text files stored on the client computer and they are kept of use tracking purpose. Server script sends a set of cookies to the browser. For example name, age, or identification number etc. The browser stores this information on a local machine for future use.
- When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user.

How to use Cookies

- How to create a cookie?

```
HttpCookie userInfo = new HttpCookie("userInfo");
userInfo["UserName"] = "Annathurai";
userInfo["UserColor"] = "Black";
userInfo.Expires.Add(new TimeSpan(0, 1, 0));
Response.Cookies.Add(userInfo);
```

How to use Cookies

- How to retrieve from cookie?

```
string User_name = string.Empty;
string User_color = string.Empty;
HttpCookie reqCookies = Request.Cookies["userInfo"];
if (reqCookies != null)
{
    User_name = reqCookies["UserName"].ToString();
    User_color = reqCookies["UserColor"].ToString();
}
```

Difference table between Cookies and Session

Session	Cookies
A session stores the variables and their values within a file in a temporary directory on the server.	Cookies are stored on the user's computer as a text file.
The session ends when the user logout from the application or closes his web browser.	Cookies end on the lifetime set by the user.
It can store an unlimited amount of data.	It can store only limited data.
We can store as much data as we want within a session, but there is a maximum memory limit, which a script can use at one time, and it is 128 MB.	The maximum size of the browser's cookies is 4 KB.
We need to call the session_start() function to start the session.	We don't need to call a function to start a cookie as it is stored within the local computer.

ASP.NET (C#)

Article 22

In this article, I'll explain an easy but an important concept of how to use **Authentication** in ASP.NET.

Models/Account

```
[Table("Account")]
public class Account
{
        [DatabaseGenerated(DatabaseGeneratedOption.None)]
        public int ID { get; set; }

        [Required(ErrorMessage = "Email is required")]
        [StringLength(100)]
        public string Email { get; set; }

        [Required(ErrorMessage = "Password is required")]
        [DataType(DataType.Password)]
        [Display(Name = "password")]
        [StringLength(50)]
        public string Password { get; set; }

}
```

HMSContext

```
public class HMSContext : DbContext
{
        public HMSContext() : base("Name=dbContext")
        {
            Database.SetInitializer<HMSContext>(new
              DropCreateDatabaseIfModelChanges<HMSContext>());
        }
        public DbSet<Product> Products { get; set; }
        public virtual DbSet<Account> Accounts { get; set; }
```

```
public class HMSContext : DbContext{
. . .
protected override void OnModelCreating(DbModelBuilder modelBuilder){
            modelBuilder.Entity<Account>()
                .Property(e => e.Email)
                .IsFixedLength();
            modelBuilder.Entity<Account>()
                .Property(e => e.Password)
                .IsFixedLength();
            modelBuilder.Entity<Product>()
                .Property(e => e.Image)
                .IsFixedLength();
        }
}
```

LoginController

```
public class LoginController : Controller{
            HMSContext db = new HMSContext ();
        public ActionResult Index(){
            return View();
        }

        [HttpPost]
        public ActionResult Index(Account acc, string returnUrl){
            if(ModelState.IsValid){
          string password = GetPassWord(acc.Email);
                if(string.IsNullOrEmpty(password)){
                    ModelState.AddModelError("", "The user login or password provided is
incorrect.");
                }
                else{
                    if(acc.Password.Equals(password.Trim())){
                        FormsAuthentication.RedirectFromLoginPage(acc.Email, false);

                    }
                    else{
                        ModelState.AddModelError("", "The password provided is
incorrect.");
                    }
                }
            }
            return View(acc);
        }
```

```
[Authorize]
public ActionResult Logout() {
            FormsAuthentication.SignOut();
            return RedirectToAction("Index", "Home");
}
public bool CheckLoginExist(string LoginName) {
            return db.Accounts.Where(x =>
                        x.Email.Equals(LoginName)).Any();
}

public string GetPassWord(string LoginName) {
            var user= db.Accounts.Where(y =>
                        y.Email.ToLower().Equals(LoginName));
            if (user.Any()){
                return user.FirstOrDefault().Password;
            }
            else
                return string.Empty;
}
```

Views/Login

```
@model HayMuaSi.Models.Account
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
    <title>login</title>
    <link href="~/Content/Login.css" rel="stylesheet" />
    <link href="~/Content/bootstrap.css" rel="stylesheet" />
    <script src="~/Scripts/jquery-1.10.2.js"></script>
    <script src="~/Scripts/bootstrap.min.js"></script>
</head>
```

```
<body>
    <div id="form-login">
        <div id="login-banner">
            <span>Login Form</span>
        </div>
        <div id="login-content">
            @using (Html.BeginForm())
            {
                @Html.AntiForgeryToken()
                    <div class="form-horizontal">
                        @Html.ValidationSummary(true, "", new { @class =
"text-danger" })
                        <div class="form-group">
                            <label class="control-label col-sm-2"
for="email">Email:</label>
                                <div class="col-sm-10">
                                 @Html.EditorFor(model => model.Email.
new { htmlAttributes = new { @class = "form-control", @id = "email",
@plancehoder = "Enter email", @type = "email" } })
                                 @Html.ValidationMessageFor(model =>
model.Email, "", new { @class = "text-danger" })
                                </div>
                        </div>
```

```
<div class="form-group">
        <label class="control-label col-sm-2"
                for="pwd">Password:</label>
        <div class="col-sm-10">
        @Html.EditorFor(model => model.Password, new { htmlAttributes
= new { @class = "form-control", @id = "pwd", @plancehoder = "Enter
password", @type = "password" } })
        @Html.ValidationMessageFor(model => model.Password, "", new {
@class = "text-danger" })
        </div>
</div>
<div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
                <div class="checkbox">
                        <label><input type="checkbox">
                                        Remember me</label>
                </div>
        </div>
</div>
```

```
<div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
                <button type="submit" class="btn btn-default">Submit</button>
        </div>
</div>
<div id="more-login">
        <div id="login-facebook">
                <a href="#">Facebook</a>
        </div>
        <div id="login-googleplus">
                <a href="#"> Google</a>
        </div>
</div>
</div>
}
<script type="text/javascript">
        $(document).ready(function () {
        $("#form-login").fadeIn("1000");
});
</script>
</div>
</div>
</body>
</html>
```