Using Areas in Asp.Net MVC with Example

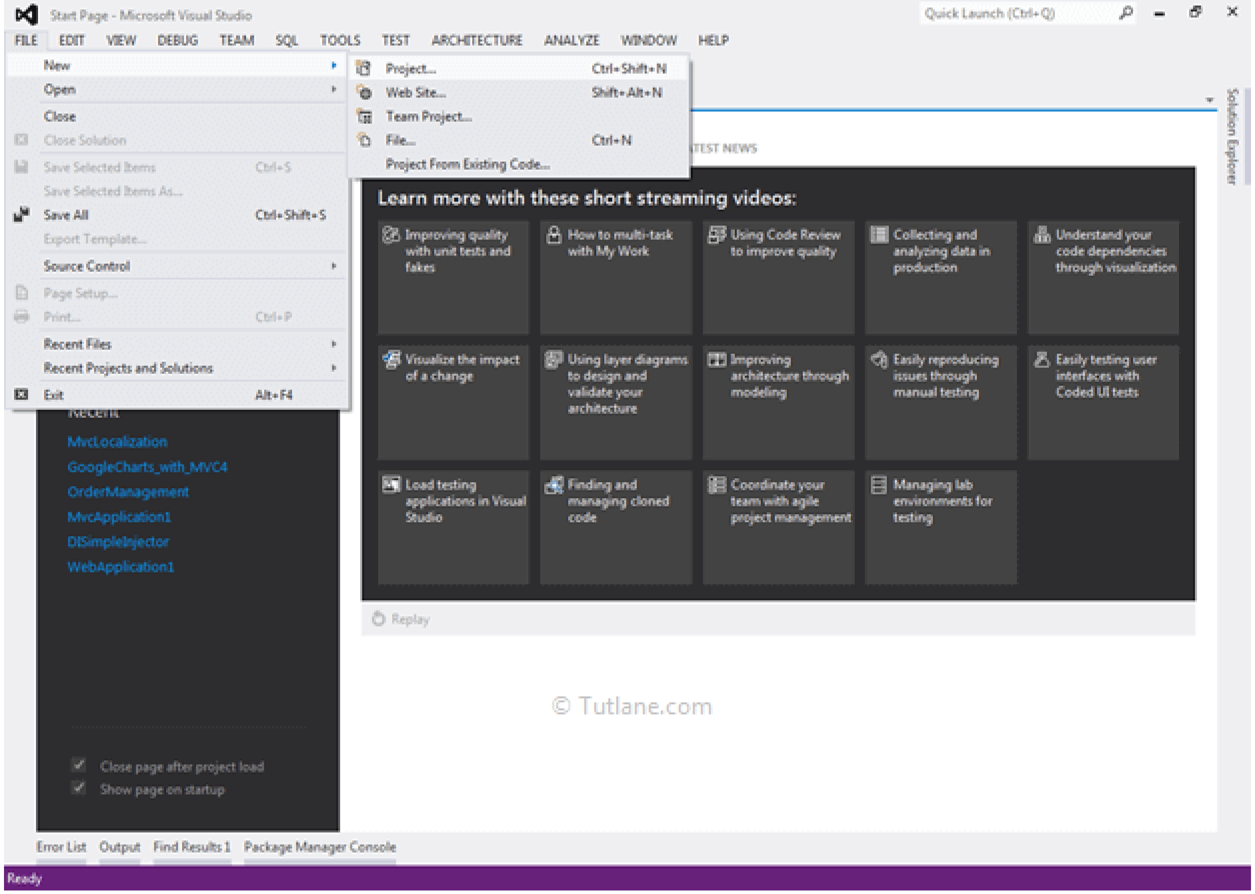
Over 1,300 marketplaces are built with Multi-Vendor worldwide
cs.cart
Find Out Why

Here we will learn areas in asp.net mvc and how to use areas in asp.net mvc with simple example. In asp.net mvc when we partition web applications into smaller units that are referred as areas. Areas provide a way to separate a large mvc web application into smaller functional groupings.

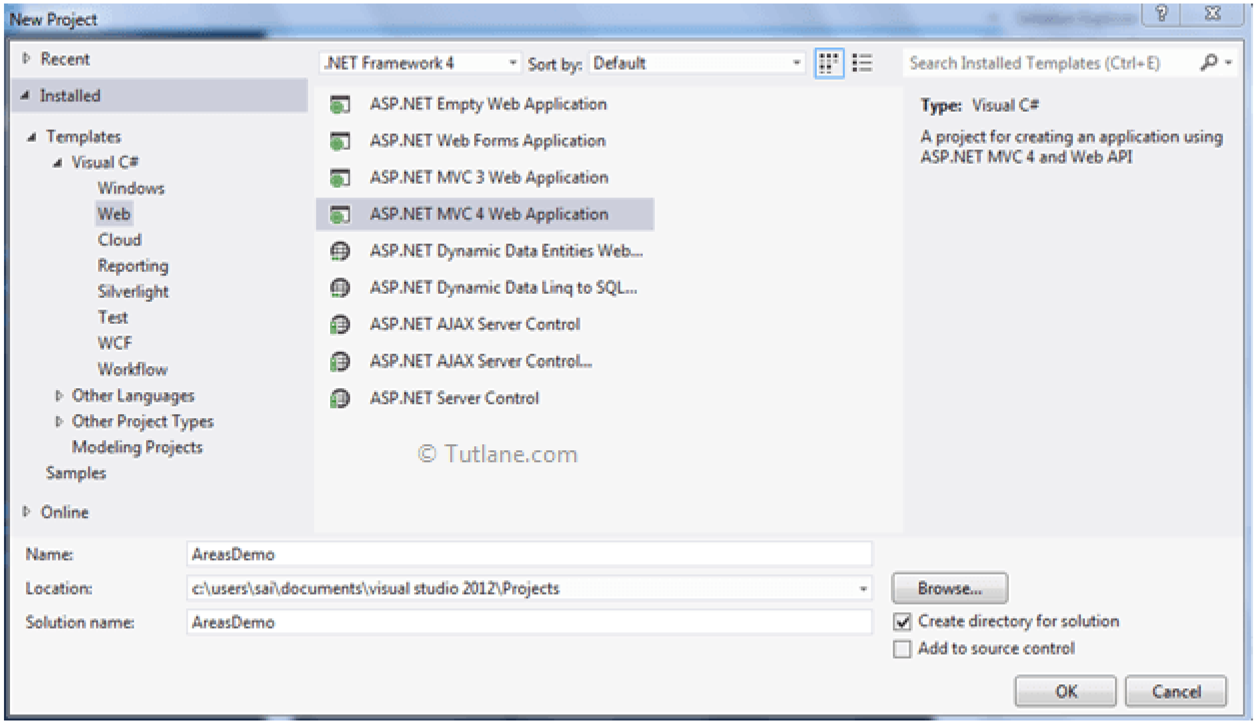
If we have small web application then the default folder structure work fine to maintain application. But if your application becomes big then it create problem to maintain for example if we have large hospital portal then there are various section in it and if we use default folder structure which we have then it will create a mess. In this condition we need to use Areas for separating our application. Now we will learn areas in asp.net with simple application

Create New Applicaiton in Asp.Net MVC

Now let's create a basic ASP.NET MVC 4 application to under stand areas in asp.net for that Open visual studio studio → Go to File → Select New → Select Project


Start Page - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Source Control
Page Setup...
Print... Ctrl+P
Recent Files
Recent Projects and Solutions
Exit Alt+F4
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
Team Project...
File... Ctrl+N
Project From Existing Code...
Learn more with these short streaming videos:
Improving quality with unit tests and fakes
How to multi-task with My Work
Using Code Review to improve quality
Collecting and analyzing data in production
Understand your code dependencies through visualization
Visualize the impact of a change
Using layer diagrams to design and validate your architecture
Improving architecture through modeling
Easily reproducing issues through manual testing
Easily testing user interfaces with Coded UI tests
Load testing applications in Visual Studio
Finding and managing cloned code
Coordinate your team with agile project management
Managing lab environments for testing
Replay
Recent
MvcLocalization
GoogleCharts_with_MVC4
OrderManagement
MvcApplication1
DISimpleInjector
WebApplication1
Close page after project load
Show page on startup
© Tutlane.com
Error List Output Find Results 1 Package Manager Console
Solution Explorer
Ready

After that you will see new dialog will pop up for selecting your Template and Project type. From Templates select Visual C# → inside that select Web and then project type select ASP.NET MVC 4 Web Application and here we are giving name as "**AreasDemo**" then finally click on OK button.


New Project
Recent
Installed
Templates
Visual C#
Windows
Web
Cloud
Reporting
Silverlight
Test
WCF
Workflow
Other Languages
Other Project Types
Modeling Projects
Samples
Online
ASP.NET Empty Web Application
ASP.NET Web Forms Application
ASP.NET MVC 3 Web Application
ASP.NET MVC 4 Web Application
ASP.NET Dynamic Data Entities Web...
ASP.NET Dynamic Data Linq to SQL...
ASP.NET AJAX Server Control
ASP.NET AJAX Server Control...
ASP.NET Server Control
© Tutlane.com
Search Installed Templates (Ctrl+E)
Browse...
OK
Cancel

.NET Framework 4 Sort by: Default

Type: Visual C#

A project for creating an application using ASP.NET MVC 4 and Web API

Name: AreasDemo

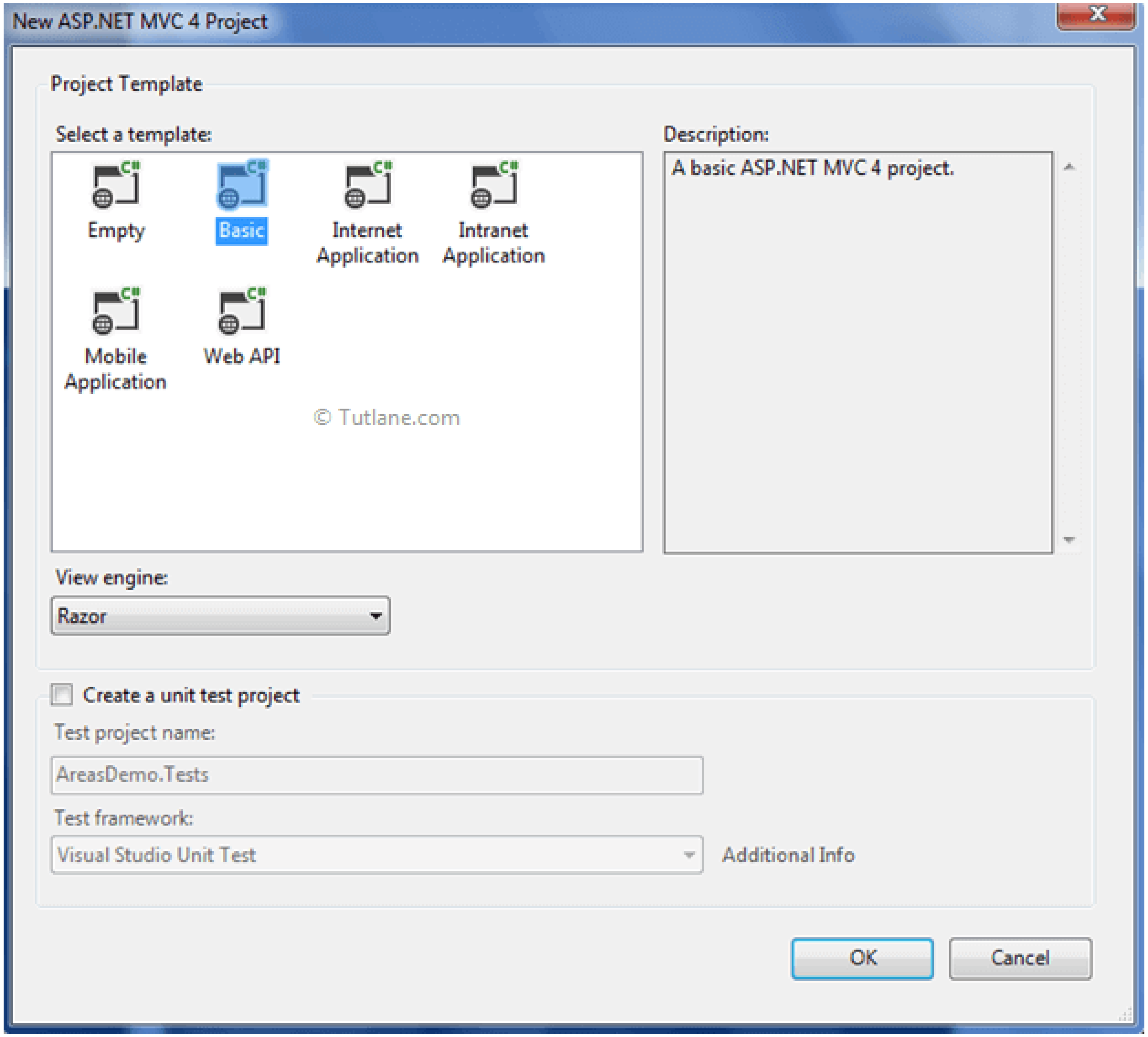
Location: c:\users\sai\documents\visual studio 2012\Projects

Solution name: AreasDemo

☑ Create directory for solution

□ Add to source control

Now new dialog will pop up for selecting template in that Select Basic Template and click ok it like as shown below



A basic ASP.NET MVC 4 project.

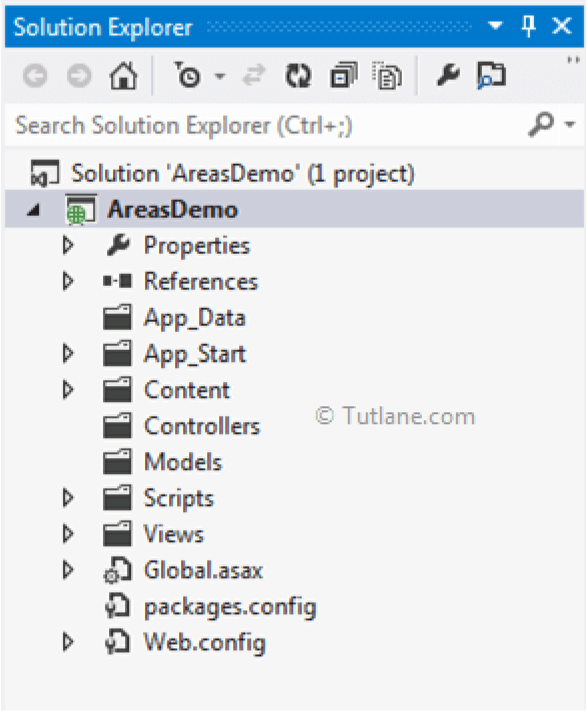
View engine: Razor

□ Create a unit test project

Test project name: AreasDemo.Tests

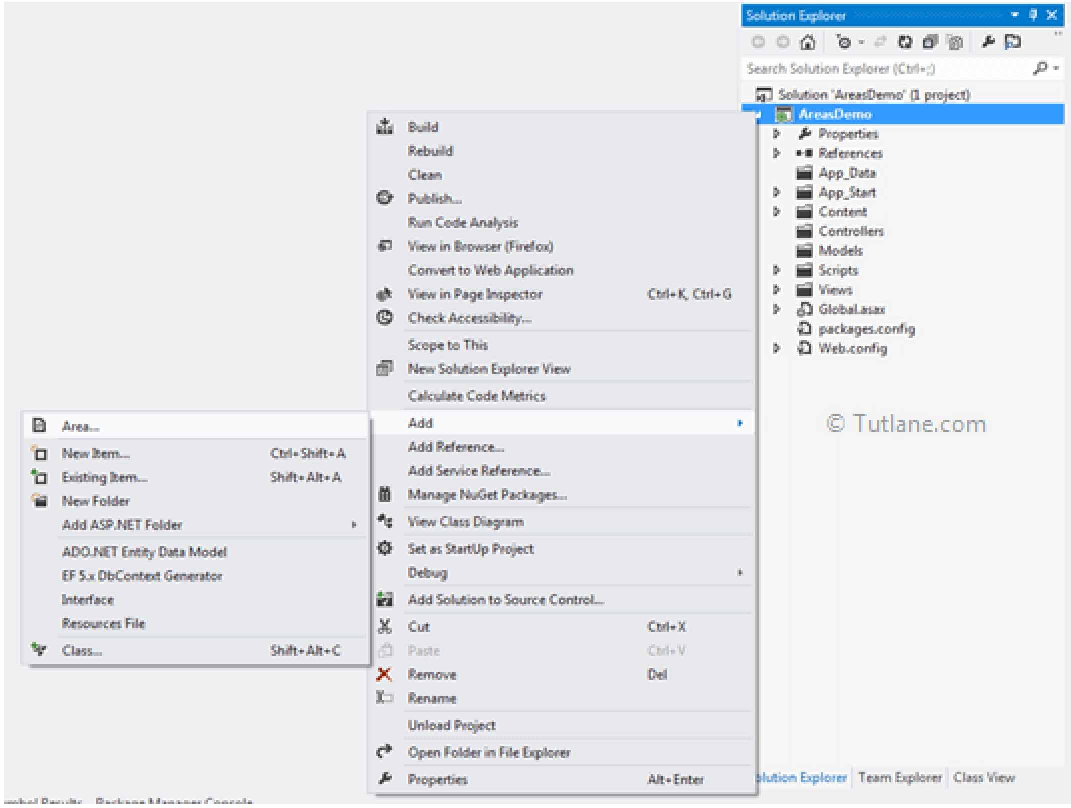
Test framework: Visual Studio Unit Test

After clicking on OK button a simple asp.net mvc project will be created that will be like as shown below

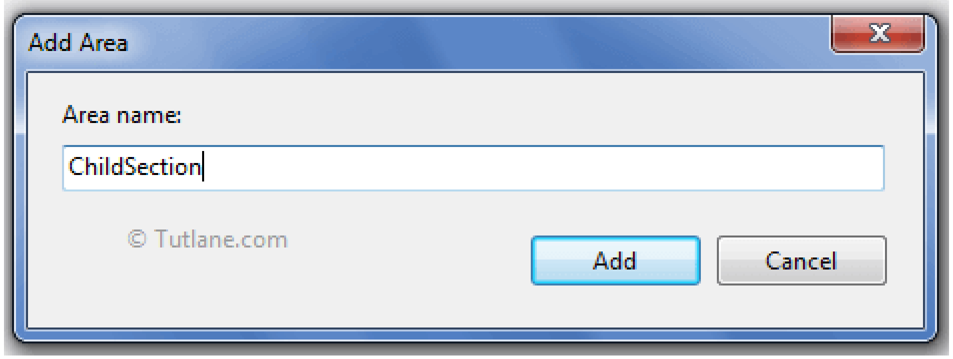


Creating Area in Asp.Net MVC Application

To add an Area to MVC application, right-click on the project item with in the Solution Explorer window and select Add → Area

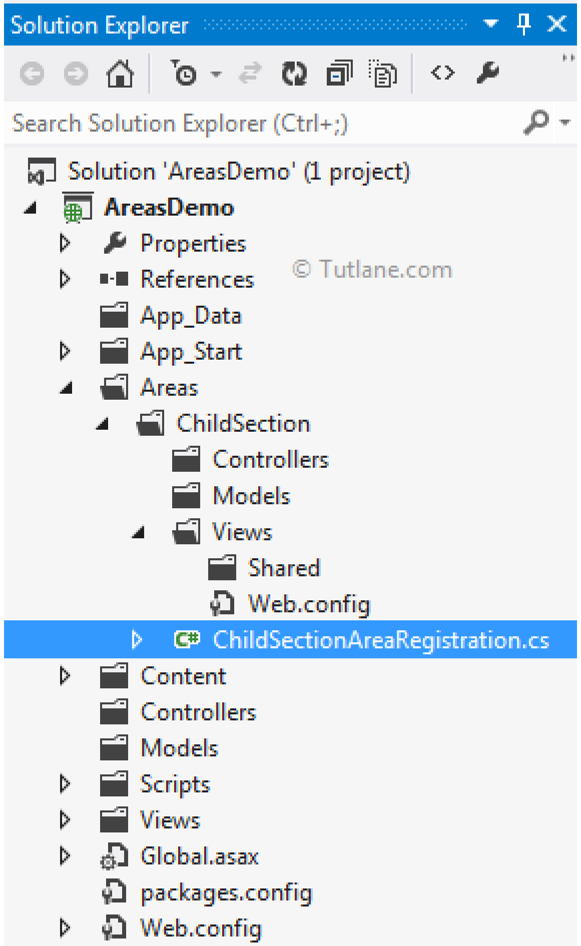


After clicking on Area a New wizard will popup for Asking Area Name in that give name as "**ChildSection**" like as shown below


Add Area

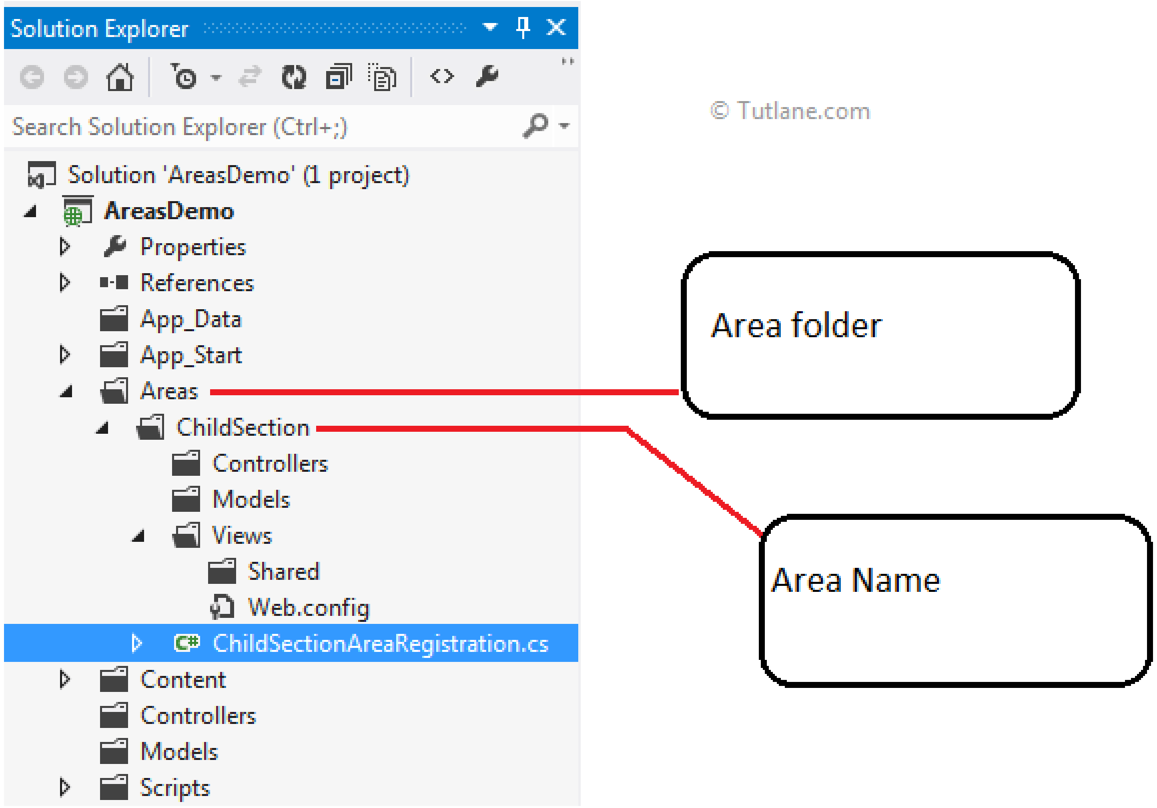
Area name: ChildSection

Now click on Add button the Area with Name ChildSection will create like as shown below



```
Solution 'AreasDemo' (1 project)
  AreasDemo
    Properties
    References
    App_Data
    App_Start
    Areas
      ChildSection
        Controllers
        Models
        Views
          Shared
          Web.config
        ChildSectionAreaRegistration.cs
    Content
    Controllers
    Models
    Scripts
    Views
    Global.asax
    packages.config
    Web.config
```

After adding if we check our folder structure we have an Areas folder created inside that we have another folder which contains Area with name ChildSection and all basic folder structure (Models/Views/Controllers) with A Web.config file under the Views folder. This contains the necessary entries for the RazorViewEngine to function properly.



If we check above folder structure we have class **ChildSectionAreaRegistration.cs** which is created inside ChildSection Area.

Registering Areas in MVC

We need to inform MVC framework that we have added Area this class is auto generated when we add Area we do not required to write it explicitly.

```
using System.Web.Mvc;
namespace AreasDemo.Areas.ChildSection
{
public class ChildSectionAreaRegistration : AreaRegistration
{
public override string AreaName
{
get
{
return"ChildSection";
}
}
public override void RegisterArea(AreaRegistrationContext context)
{
context.MapRoute(
"ChildSection_default",
"ChildSection/{controller}/{action}/{id}",
new { action = "Index", id = UrlParameter.Optional }
);
}
}
}
```

This ChildSectionAreaRegistration class is inheriting from Abstract class AreaRegistration which has two method (**AreaName**, **RegisterArea**) as abstract which we are overriding in this ChildSectionAreaRegistration Class. The first method AreaName contains name of your Area (ChildSection) and the second method contains Routing for ChildSection Area.

Registering Areas in Global.asax

In Global.asax we need to register all Areas in Application_Start event handler by adding below code.

```
AreaRegistration.RegisterAllAreas();
```

Here RegisterAllAreas() Method is a static method. and this RegisterAllAreas() method will call all RegisterArea() methods of all Areas found in application. Once we add above code in Global.asax Application_Start event that will be like as shown below

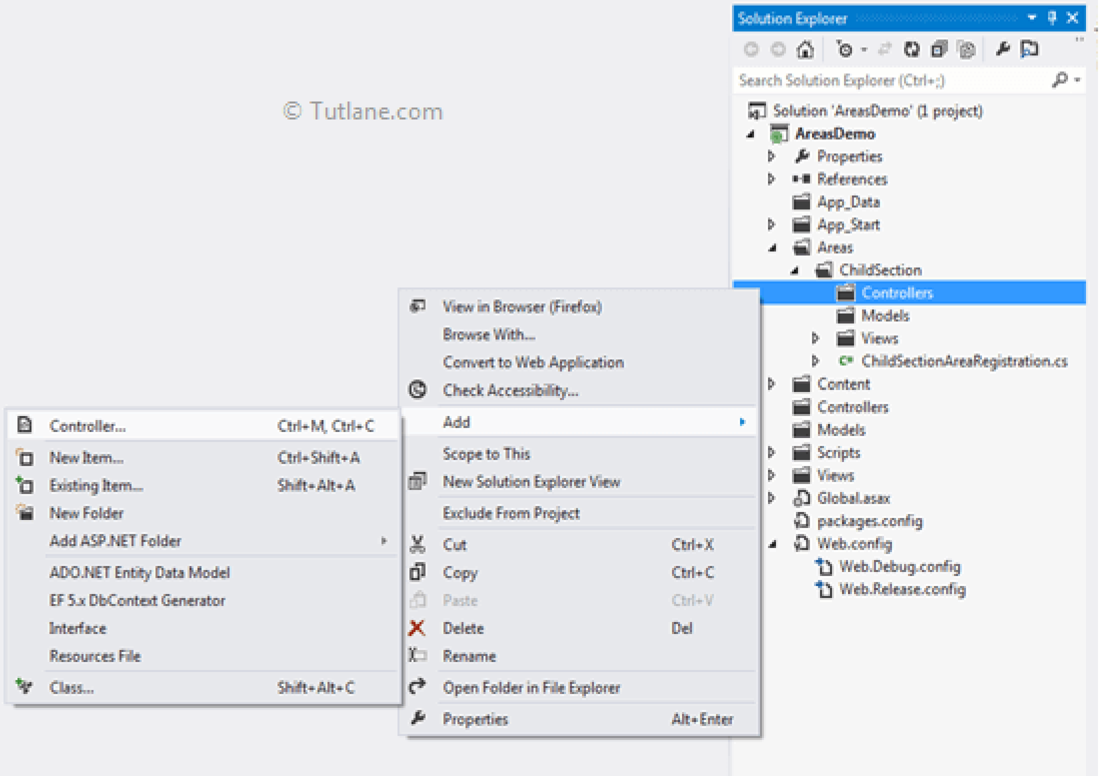
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Http;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace AreasDemo
{
// Note: For instructions on enabling IIS6 or IIS7 classic mode,
// visit http://go.microsoft.com/?LinkId=9394801
public class MvcApplication : System.Web.HttpApplication
{
protected void Application_Start()
{
AreaRegistration.RegisterAllAreas();
WebApiConfig.Register(GlobalConfiguration.Configuration);
FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
RouteConfig.RegisterRoutes(RouteTable.Routes);
BundleConfig.RegisterBundles(BundleTable.Bundles);
}
}
}
```

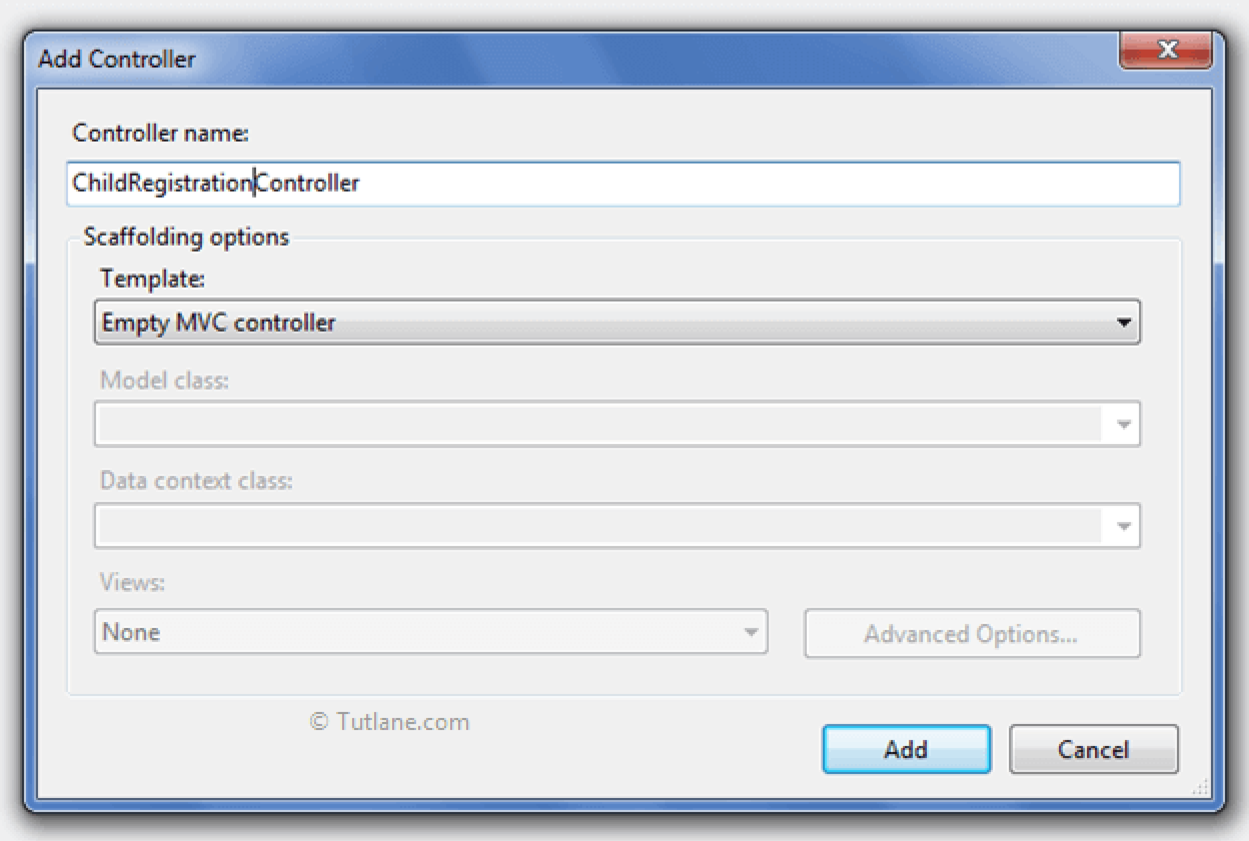
After completing registering Area now let's add Controller and View in this Area and check how its works.

Adding Controller in Asp.Net MVC

For adding controller Just Right click on Controller Folder inside that select Add and then select Controller



After clicking on controller new dialog will popup with name Add Controller in that give name to controller and select template as "Empty MVC Controller" and click on Add button.

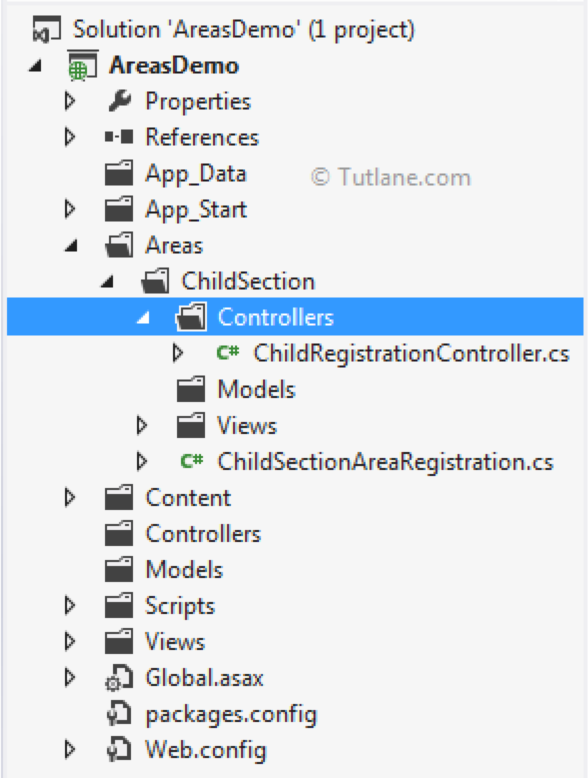


Controller name: ChildRegistrationController

Template: Empty MVC controller

None

After adding new controller our project will be like as shown below



```
Solution 'AreasDemo' (1 project)
  AreasDemo
    Properties
    References
    App_Data
    App_Start
    Areas
      ChildSection
        Controllers
          ChildRegistrationController.cs
        Models
        Views
        ChildSectionAreaRegistration.cs
    Content
    Controllers
    Models
    Scripts
    Views
    Global.asax
    packages.config
    Web.config
```

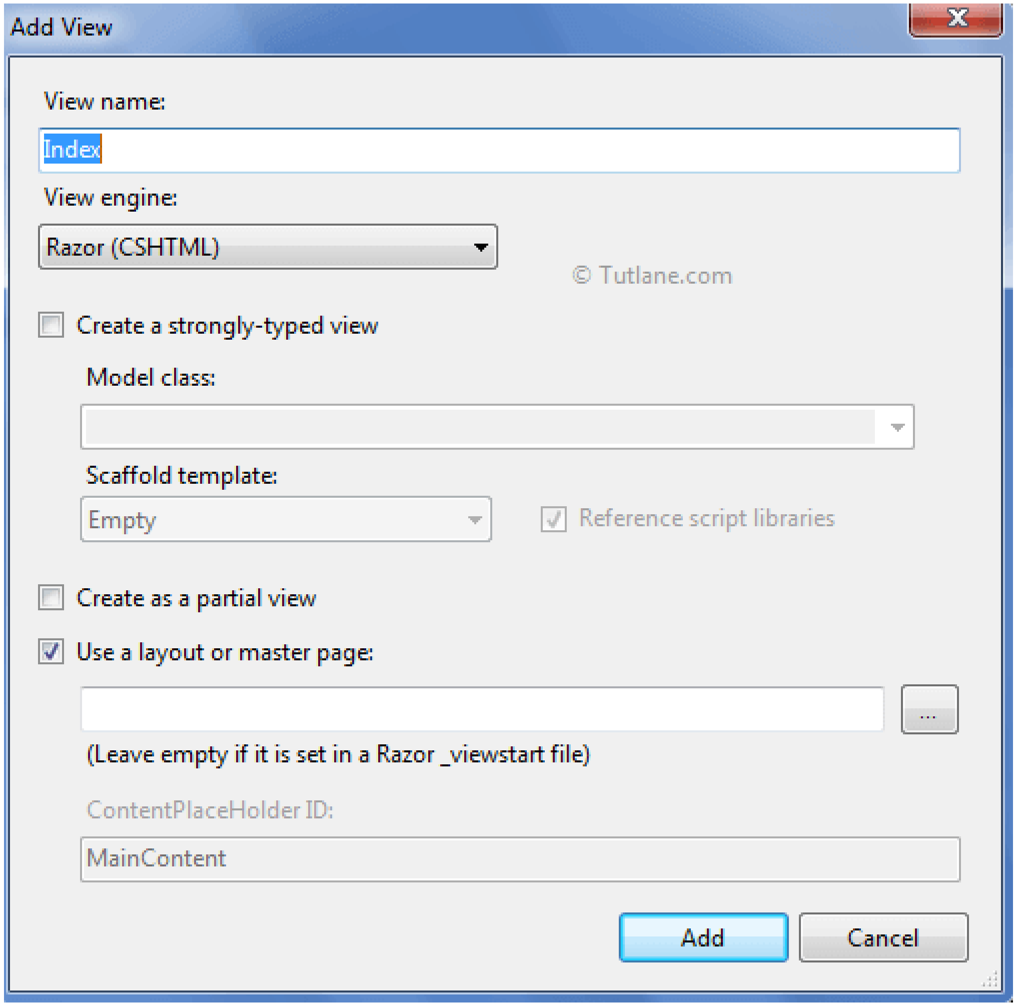
Our controller will contain code like as shown below

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

namespace AreasDemo.Areas.ChildSection.Controllers
{
public class ChildRegistrationController : Controller
{
//
// GET: /ChildSection/ChildRegistration/
public ActionResult Index()
{
return View();
}
}
}
```

Adding View in Asp.Net MVC Application

For Adding View just right click inside Index ActionResult Method and Select "Add View" to create the view template for our Index form. Now new pop up "Add View" will open in that give view name as "Index" and click Add button (Here we are not changing any option in wizard) like as shown.



Add View

View name:

Index

View engine:

Razor (CSHTML)

© Tutlane.com

□ Create a strongly-typed view

Model class:

Scaffold template:

Empty

☑ Reference script libraries

□ Create as a partial view

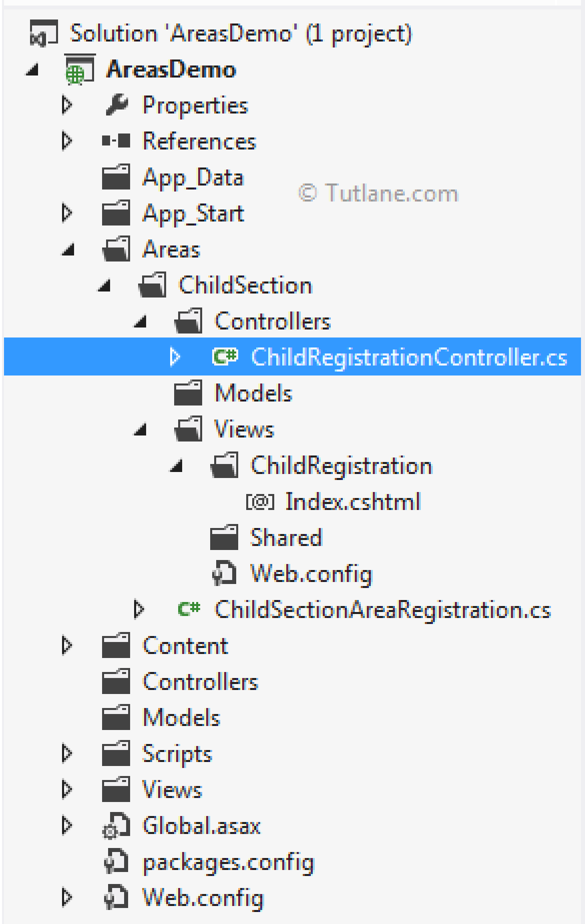
☑ Use a layout or master page:

(Leave empty if it is set in a Razor _viewstart file)

ContentPlaceHolder ID:

MainContent

After adding view our project structure will be like as shown below



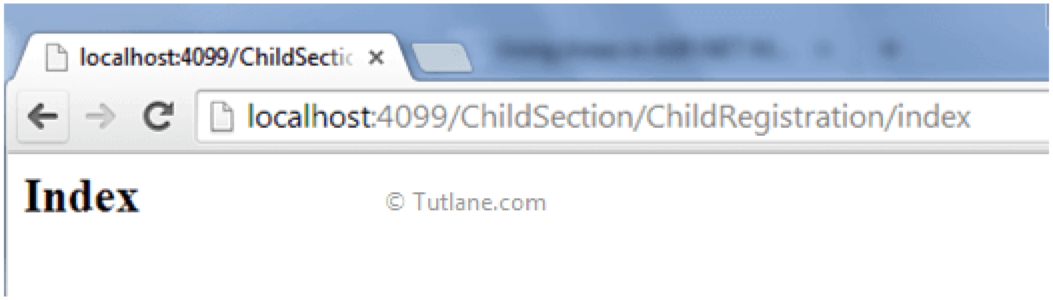
Now run your application and see the output.

Accessing URL in Asp.Net MVC

For accessing URL we need to enter Area Name first then Controller Name and lastly Action name.

Area Name - ChildSection.
Controller Name - ChildRegistration
Action Name - Index

Our url will be like as **http://localhost:4099/ChildSection/ChildRegistration/Index** and our output will be like as shown below

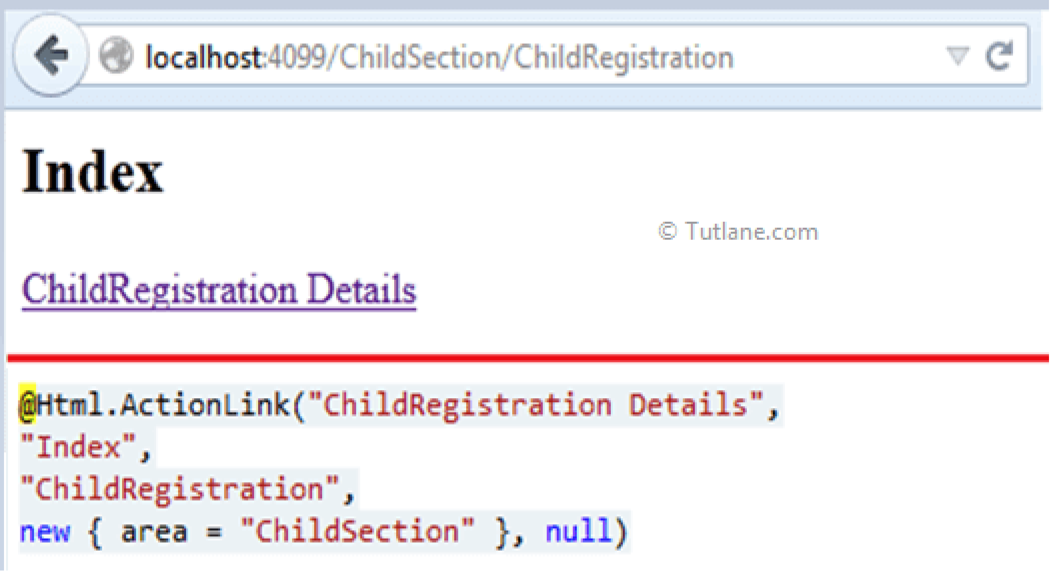

localhost:4099/ChildSectic ×
localhost:4099/ChildSection/ChildRegistration/index
© Tutlane.com

Index

Here we will see how to create link with area in our view. We need to write action link with area after controller name just add New { area= “Area Name”} like as shown below.

```
@Html.ActionLink("ChildRegistration Details", "Index", "ChildRegistration", new { area = "ChildSection" }, null
)
```

Our output will be like as shown below



Index

ChildRegistration Details

```
@Html.ActionLink("ChildRegistration Details",
"Index",
"ChildRegistration",
new { area = "ChildSection" }, null)
```

CONTACT US

Address: No.1-93, Pochamma Colony, Manikonda, Hyderabad, Telangana - 500089

✉ **Email:** support@tutlane.com (mailto:support@tutlane.com)