



Introduction

Hi guys! This article explains how to create areas in asp.net core 3.1. We will make multiple areas and call them from the views. You'll also learn to set the routing of areas.

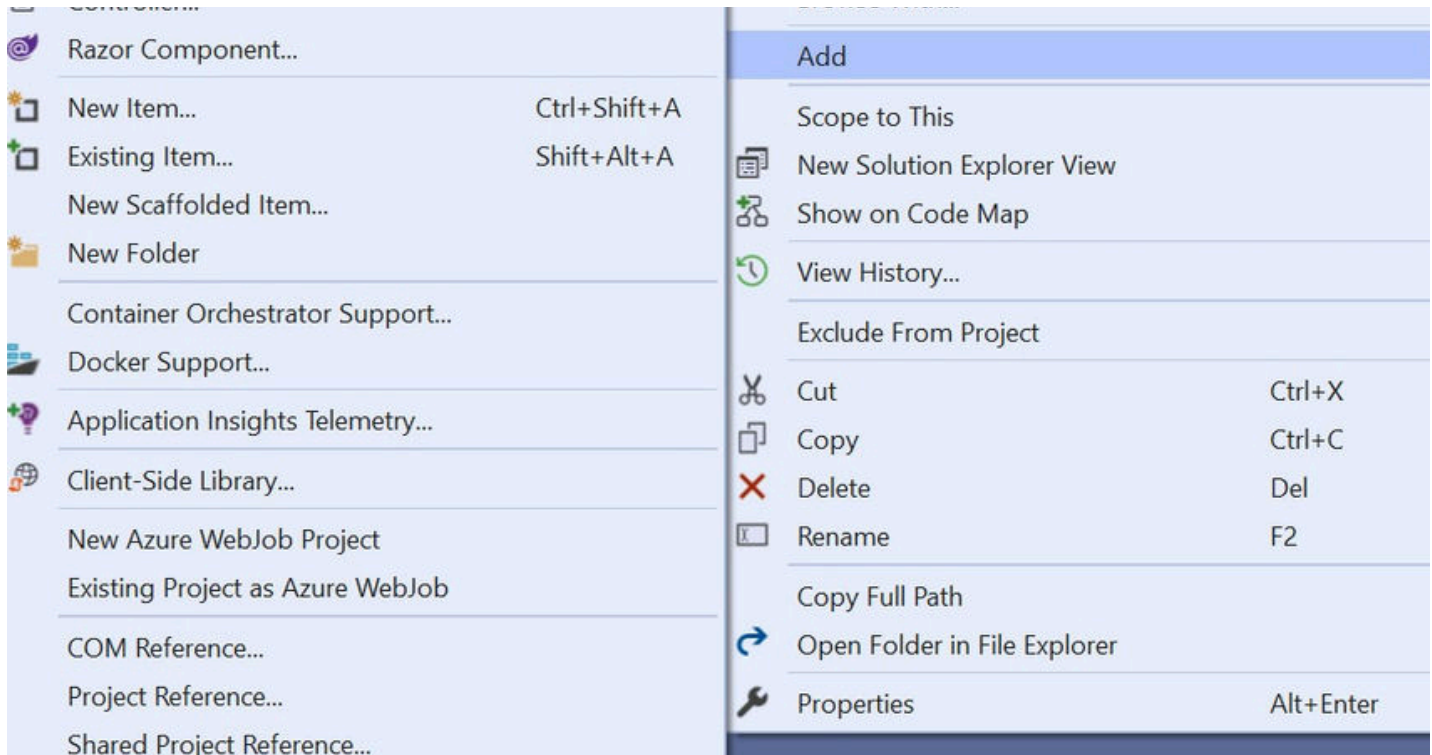
What are Areas?

- Areas are part of MVC, which is a separate module in our application.
- We can create a level application by diving the modules areawise.
- We can create the same name controller and view it multiple times.
- We can create areas for Employees, Admin, Customers, etc.
- There is a shared folder like the MVC, which contains the `_Layout` of the application. and `_ViewStart`, and `_ViewImport` views in the views folder.

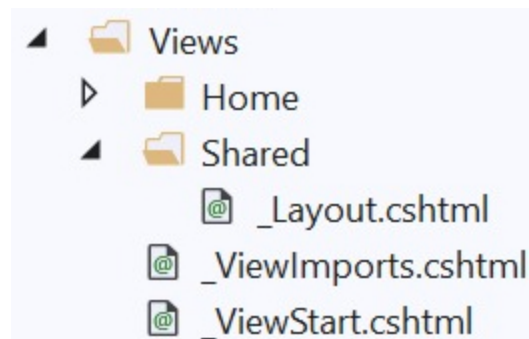
While executing any view, the `_Layout` is called, but before it, `_ViewStart` and `_ViewImport` are called. `_ViewStart` contains the "`_Layout`" path that is auto-applied to every view page without adding it to every view. `_ViewImport` contains all the tag helpers and the namespaces that we want to use over the view page for ignoring to add it multiple times.

Okay, let's see how to create it.

Step 1. Right-click on Areas folder > Add > Area> Give it name Admin >ok



Step 2. In .net core, the Shared folder is not auto-created so, we have to create it manually by copying the Shared folder in the Admin Areas Views folder.



Step 3. Because Area is the MVC concept, for enabling areas in .Net Core, we have to register MVC. And EnableEndpointRouting is set to false in the startup.cs file.

```
1 services.AddMvc(options => options.EnableEndpointRouting = false);
```

.Net Core auto-registered controllers with views and the razor pages, but if it is not registered, in your web application, in any case, please register it like this.

```
1 services.AddControllersWithViews();
2 services.AddRazorPages();
```

Step 4. Now we set the routing of areas so we can call it.

Embed the following tag in the Configure method for setting the route of the area in your startup.cs file.

```
1 endpoints.MapAreaControllerRoute(  
2     name: "Admin",  
3     areaName: "Admin",  
4     pattern: "Admin/{controller=Home}/{action=Index}"  
5 );
```

```
1 endpoints.MapAreaControllerRoute(  
2     name: "Customer",  
3     areaName: "Customer",  
4     pattern: "Customer/{controller=Home}/{action=Index}"  
5 );
```

Here is the full configuration method with default routing and Admin area routing.

```
1 app.UseEndpoints(endpoints =>  
2 {  
3     endpoints.MapAreaControllerRoute(  
4         name: "Admin",  
5         areaName: "Admin",  
6         pattern: "Admin/{controller=Home}/{action=Index}"  
7     );  
8  
9     endpoints.MapControllerRoute(  
10        name: "default",  
11        pattern: "{controller=Home}/{action=Index}/{id?}"  
12    );  
13  
14    endpoints.MapRazorPages();  
15 });
```

Note. Make sure you embed any area routing before default routing.

Step 5. Okay, now I created the Home controller and its index method in the Admin area and the Customer Area.

.Net core assumes every controller is a simple controller. So, to identify each area's controller, we will decorate the area's controller with the `[Area("admin")]` attribute.

```
3 {  
4     public IActionResult Index()  
5     {  
6         return View();  
7     }  
8 }
```

```
1 [Area("customer")]  
2 public class HomeController : Controller  
3 {  
4     public IActionResult Index()  
5     {  
6         return View();  
7     }  
8 }
```

Step 6. Let us see how to call any area from the view.

Way 1

```
1 <a asp-area="admin" asp-controller="Home" asp-action="index">Go to Admin
```

```
1 <a asp-area="customer" asp-controller="Home" asp-action="index">Go to Cus
```

Way 2

```
1 <a href="@Url.Action("index","home",new {Area="admin"})">Go to Admin Area
```

```
1 <a href="@Url.Action("index","home",new {Area="customer"})">Go to Custome
```

Way 3

```
1 <a href="/admin/Home/index">Go to Admin Area</a>
```

```
1 <a href="/customer/Home/index">Go to Customer Area</a>
```

Call the front index view like this

```
1 <a asp-action="index" asp-controller="home">Go to Home</a>
```

Go to Home

[Go to Admin Area](#)
[Go to Customer Area](#)

← → ↻ 🔒 localhost:44313/Admin

[Admin Area](#) [Home](#) [Privacy](#)

Admin Area Index

[Customer Area](#) [Home](#) [Privacy](#)

Customer Area Index

Summary

In this article, I discussed how we create areas in the .net core so that we can separate modules in the application. Furthermore, we saw how to work with multiple areas and set the routing. At last, we saw the calling of two different areas from the view.

[ASP.NET Core](#)[MVC](#)[Areas](#)[Routing](#)[Shared Folder](#)[Controller](#)[View](#)

RECOMMENDED FREE EBOOK

[Download Now!](#)

SIMILAR ARTICLES

[Bilingual with .Net Core MVC or Globalization And Localization In ASP.NET Core MVC 3.1](#)[Custom Login And Register With Identity In ASP.NET Core 3.1](#)[Upload And Display Image In ASP.NET Core 3.1](#)[Getting Started With ASP.NET Core 3.1 - Part Two](#)[Model Validation In ASP.NET MVC Core 3.1](#)

Khushbu Saini *TOP 500*

Passionate Web developer building excellent Websites , 3 years of experience in Website Development especially in .Net Core , MVC , JQuery & Angular Technology . Team leader of Web Projects , and a Technical blog Wr... [Read more](#)

462

1.3m

1

[View All Comments](#)

3



Type your comment here and press Enter Key (Minimum 10 characters)

Follow Comments

[About Us](#) [Contact Us](#) [Privacy Policy](#) [Terms](#) [Media Kit](#) [Sitemap](#) [Report a Bug](#) [FAQ](#) [Partners](#)

[C# Tutorials](#) [Common Interview Questions](#) [Stories](#) [Consultants](#) [Ideas](#) [Certifications](#) [Sharp TV](#)

[Web3 Universe](#) [Build with JavaScript](#) [Let's React](#) [DB Talks](#) [Jumpstart Blockchain](#) [Interviews.help](#)

©2024 C# Corner. All contents are copyright of their authors.

