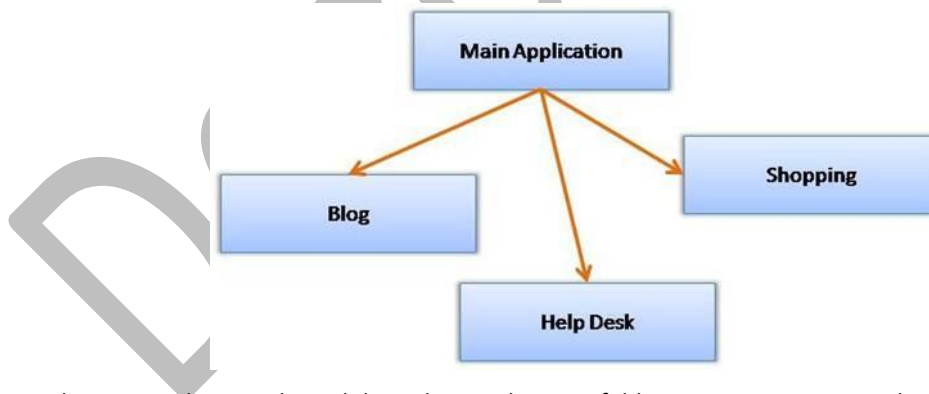


Agenda: Working with Areas

- Understanding Areas
- Adding Areas
- Registering Areas
- Linking between Areas

Understanding Areas

- The MVC pattern separates the model (data) logic of an application from its presentation logic and business logic. In ASP.NET MVC, this logical separation is also implemented physically in the project structure, where controllers and views are kept in folders that use naming conventions to define relationships. This structure supports the needs of most Web applications.
- However, some applications can have a large number of controllers, and each controller can be associated with several views. For these types of applications, the default ASP.NET MVC project structure can become complicated to maintain.
- To accommodate large projects, ASP.NET MVC lets you partition Web applications into smaller units that are referred to as areas. Areas provide a way to separate a large MVC Web application into smaller functional groupings. An area is effectively an MVC structure inside an application. An application could contain several MVC structures (areas).
- For example, a single large application might be divided into areas that represent the Blog, HelpDesk, and Shopping. Each area represents a separate function of the overall application.



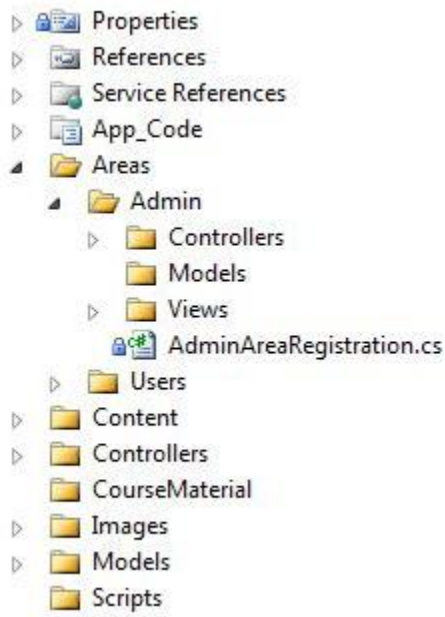
Note: With areas in place, each module replicates the MVC folder structure. For example, each module will have its own Models, Views and Controllers folder. You can then have HomeController class in Blog module as well as HelpDesk module. So, in the above example, essentially you will have four MVC structures - one for the main application and one each for the three modules (Blog, HelpDesk and Shopping).

Adding Area

In Solution Explorer, right-click the project name, click Add, and then click Area.

In Area Name, type Admin and then click Add.

An Areas folder is added to the project. The Areas folder contains a folder structure that allows each child area to have its own models, views, and controllers.



Adding Area-Specific Controllers:

1. In Solution Explorer, right-click the Controllers subfolder for the Admin area, click Add, and then click Controller.
2. Name the controller HomeController and then click Add.

Registering Areas

When you add an area to a project, a route for the area is defined in an **AreaRegistration** file. The route sends requests to the area based on the request URL. To register routes for areas, you add code to the Global.asax file that can automatically find the area routes in the AreaRegistration file.

Following line will be present in Application_Start in Global.asax

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

This code calls the route registration methods for each child area.

Linking between Areas

ASP.NET MVC Views often need to link to some action method residing in controller classes. Unless otherwise specified, action methods and controllers are assumed to be from the current area. In case you wish to link to an action method belonging to some other area, here is how you would do just that:

Using ActionLink:

```
@Html.ActionLink("Admin Area", "Index", "Home", new { area = "Admin" }, null)
```

Using RedirectToAction:

```
RedirectToAction("Index", "Home", new { Area = "Admin" });
```