**Understanding Attribute Routing in ASP.Net MVC**

ASP.NET MVC 5 has a new feature called Attribute Routing that allows a developer to specify the route of controller actions by adding Route Attributes to them. Developers can also add a RoutePrefix Attribute on the controller if each of the controller actions within that controller share a particular route prefix.

Article below will demonstrate the usage of Attribute Routing with sample application.

**Enable Attribute Routing**

If you want to use Attribute Routing, you have to enable it by calling MapMvcAttributeRoutes on the RouteCollection for your app (usually this is done in RouteConfig):

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapMvcAttributeRoutes(); //Enables Attribute Routing

}

}

Following are some examples of specifying attribute routing

* [RoutePrefix("Users")]
* [Route("~/")] //Specifies that this is the default action for the entire application. Route: /
* [Route("")] //Specifies that this is the default action for this route prefix. Route: /Users
* [Route("{action=index}")] //Specifies the Index action as default for this route prefix
* [Route("Users/Index")]
* [Route("Users/{id}")]
* [Route("Users/{id}/{name?}")]
* [Route("Users/{id:int}")]

Attached is the sample application to give you better idea how code migration works.

Reference:

<http://www.asp.net/mvc/overview/getting-started/getting-started-with-ef-using-mvc/migrations-and-deployment-with-the-entity-framework-in-an-asp-net-mvc-application>

<http://www.entityframeworktutorial.net/code-first/code-based-migration-in-code-first.aspx>

EF,Entity Framework, code migration,Code First