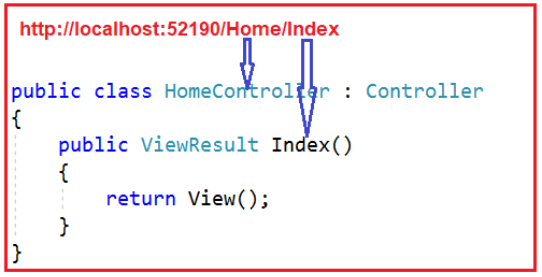
* Routing module is responsible for mapping the incoming browser requests (i.e. the incoming URL) to a particular MVC controller action method
* This mapping is done by the routing rules defined for your application
* For example, if we issue a request to the **“/Home/Index**” URL, then it is the Index action method of Home Controller class which is going to handle the request as shown in the below image.



**Types**

* Convention Based Routing
* Attribute-Based Routing.

**Convention based routing** is like URL Pattern and Custom routing

**URL Pattern in ASP.NET MVC Routing**

* Suppose your web application is running on www.dotnettutorials.net then the URL pattern **“{controller}/{action}/{id}”**  for your application would be look like [www.dotnettutorials.net/{controller}/{action}/{id}](http://www.dotnettutorials.net/%7bcontroller%7d/%7baction%7d/%7bid%7d).
* Anything after the “**www.dotnettutorials.net/**” would be considered as the controller name.
* The same way, anything after the controller name would be considered as the action name and the value of the id parameter

**Custom Routing**

* You can configure your custom route using the MapRoute extension method. You need to provide at least two parameters in MapRoute function i.e. route name and URL pattern
* You can register multiple custom routes with different names. Consider the following example where we register the “Employee” route.

public static void RegisterRoutes(RouteCollection routes)

{

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

routes.MapRoute(

name: "Employee",

url: " Employee/{id}",

defaults: new { controller = "Employee", action = "Index" }

);

routes.MapRoute(

name: "Default", //Route Name

url: "{controller}/{action}/{id}", //Route Pattern

defaults: new

{

controller = "Home", //Controller Name

action = "Index", //Action method Name

id = UrlParameter.Optional //Defaut value for above defined parameter

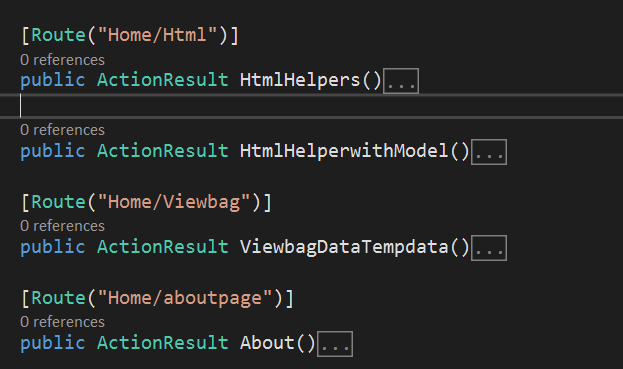
}

);

* MVC framework evaluates each route in sequence. It starts with the first configured route and if incoming URL does not satisfy with the First URL pattern of the route then it will evaluate the second route and so on.
* In the above example, the routing engine will evaluate the Employee route first and if the incoming URL does not start with domainName/Employee then only it will consider the second route which is the default route

**Attribute Routing**

* Attributes Routing are used to define the routes in controller/action directly
* The main purpose of attribute routing is to control the URLs by define directly on action and the controller in the application
* Another one purpose is no one Hack or guess the folder structure based on the URL
* For example, if user have ordering books based on authors, items based on color and so on. Then URL will look like /user/1/orders, this type of URL is difficult to match using the convention-based routing. However, it can be achieve by convention-based routing but it will take our extra effort for this. By using attribute routing, it can be done much easier, by defining on the controller/action
* Using the **[Route]** attribute to define routes is called Attribute Routing. It provides you more control over the URIs by defining routes directly on actions and controllers in your ASP.NET MVC application
* Ex: we have multiple action method methods and configured multiple action method as attribute routing means we have to manually type the URL of what you mentioned in above the action method



* <http://localhost:54020/Home/aboutpage>
* <http://localhost:54020/Home/Viewbag>

**How to Enabling Attribute Routing in ASP.NET MVC Application**

* Enabling the Attribute Routing in ASP.NET MVC Application is very simple. You just need to add a call to **routes.MapMvcAttributeRoutes()** method within **RegisterRoutes()** method of **RouteConfig.cs** file.
* So, open **RouteConfig.cs** file which is stored within the **App\_Start** folder and then just adds **routes.MapMvcAttributeRoutes();** method just above the **routes.MapRoute** as the show is below.

namespace AttributeRoutingDemoInMVC

{

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

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

//Enabling attribute routing

routes.MapMvcAttributeRoutes();

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }

);

}

}

}