**Let’s take a closer Look of**

Routing in Asp.Net MVC

Routing is a mechanism which is define that what action method is executes into controller class. Routing is a pattern matching system that monitors or checks the incoming request and takes the appropriate decisions what to do with that request. At runtime, Routing engine use the Route table for matching the incoming request's URL pattern against the URL patterns defined in the Route table. All the configured routes of an application stored in RouteTable and will be used by routing engine to determine appropriate handler class.

When the routing engine finds a match in the route table for the incoming request's URL, it forwards the request to the appropriate controller and action

ASP.NET MVC routing is a pattern matching system that is responsible for mapping incoming browser requests to specified MVC controller actions. When the ASP.NET MVC application launches then the application registers one or more patterns with the framework's route table to tell the routing engine what to do with any requests that matches those patterns

**Conditions**

1. When the request's URL matches any of the registered route patterns in the route table then the routing engine forwards the request to the appropriate handler for that request. Thereafter the route is processed and gets a view on the UI or render on the browser
2. When the request's URL does not match any of the registered route patterns then the routing engine indicates that it could not determine how to handle the request by returning a 404 HTTP status code.

**Inside App\_start there is a file called RouteConfig.cs which is use to define the multiple routes it has a single method RegisterRoute and also contain the default Routing configuration with some Default property**

**Properties of the Route**

1. RouteName: A route is mapped to a handler. A handler can be a controller in the MVC that process the request.
2. URL Pattern: A URL Pattern contain segment the controller name , action, and id, If the URL doesn't contain anything after domain name then the default controller and action method will handle the request. For example, <http://localhost:8422/> would be handled by HomeController and Index method as configured in the defaults parameter. For eg <http://localhost/home/about>

For eg url: "{controller}/{action}/{id}"

1. defaults:  The defaults is an object that contains default route value means which default controller will be execute and which default action will be execute. .

**Url Patterns means what exactly the way to action which controller and which action method**

1. {controller}/{Action}
2. Home/Index
3. {Controller}/{Action}/{ID} Note : ID is optional

**Url Pattern can be customize when we define our own routes**

We can also configure the multiple Route in a single application as per need of Application so configuration of multiple route we have to create the new route in RouteConfig.cs

Important

The register route method create the route table which is defined inside

**RouteConfig.cs** we can add our own route by calling the MapRoute method by the object of RouteCollection class. The default route table contain the single route

**User Define Route only Change the action**

routes.MapRoute(

name: "Route2",

url: "R1/",

defaults: new { controller = "Home", action = "index2", id = UrlParameter.Optional }

);

routes.MapRoute

(name:"MyRoute",

url: "Route1/",

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

);

**User Define Route change the controller and Action Method**

routes.MapRoute(

name: "SR2",

url: "SR1/",

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

);

We can close the default route and open or create the new route

routes.MapRoute(

name: "NewRoute",

url: "R1/",

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

);

//routes.MapRoute(

// name: "Default",

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

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

//);

**New Controller with New Action**

routes.MapRoute(

name: "Route2",

url: "R2/",

defaults: new { controller = "Process", action = "ActionProcess", id = UrlParameter.Optional }

);

**Default Route**

routes.MapRoute(

name: "Default",

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

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

);