

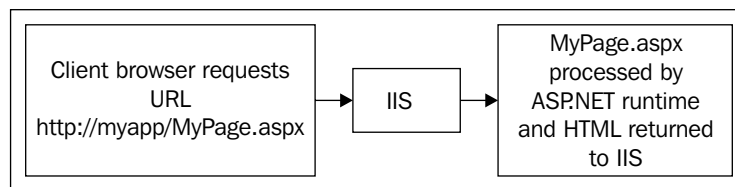
We will create a simple customer management application. For this, we first create some ASPX pages in the `Views` folder. Note that VS has already created these subfolders for us, under `Views`:

- **Home:** Contains the `Index` views
- **Shared:** Contains shared views such as master pages

Before we go on to adding custom code in this project, let us understand what VS has done for us while creating this MVC project.

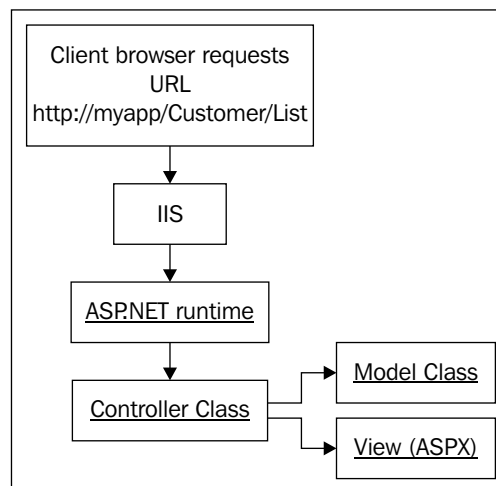
## URL Routing Engine

In the standard ASP.NET model (or Postback model), the URLs map directly to the physical files:



So when we make a request to a page, say `MyPage.aspx`, the runtime compiles that page and returns the generated HTML back to IIS to be displayed by the client browser. So we have a one-to-one relationship between the application URLs and the page.

But in the MVC framework, the URLs map to the controller classes.



Therefore, the URL is sent to IIS and then to ASP.NET runtime, where it initiates a controller class based on the URL, using the URL routes, and the controller class then loads the data from the model, with this data finally being rendered in the view.

The controller classes uses URL routing to map the URLs, which in simpler terms means rewriting URL. We can set up the rules for which URL is to be routed to which controller class. The routing will pick up the appropriate controller and pass in the query string variables as necessary.

Open the `global.asax.cs` file and examine the following code:

```
public class GlobalApplication : System.Web.HttpApplication
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
        routes.MapRoute(
            "Default",
            // Route name
            "{controller}/{action}/{id}",
            // URL with parameters
            new { controller = "Home", action = "Index", id = "" }
        )// Parameter defaults
    };
    protected void Application_Start()
    {
        RegisterRoutes(RouteTable.Routes);
    }
}
```

The `RegisterRoutes()` method contains the URL mapping routes. Initially we have only the default rule set:

```
routes.MapRoute(
    "Default",
    // Route name
    "{controller}/{action}/{id}",
    // URL with parameters
    new { controller = "Home", action = "Index", id = "" }
    // Parameter defaults
);
```

This URL mapping engine comes from `System.Web.Routing.dll`, which can be used independently, without the ASP.NET MVC framework, to rewrite URLs in your standard ASP.NET web applications.