**Setup mvc in asp.net core**

**Two steps to setup MVC in ASP.NET Core Application**   
  
**Step 1 :**In ConfigureServices() method of the Startup class in Startup.cs file, include the following line. This line of code adds the required MVC services to the dependency injection container in asp.net core.

services.AddMvc();

**Step 2 :**In the Configure() method, add UseMvcWithDefaultRoute() midddleware to our application's request processing pipeline. Modify the code as shown below.

public void Configure(IApplicationBuilder app, IHostingEnvironment env)  
{  
    if (env.IsDevelopment())  
    {  
        app.UseDeveloperExceptionPage();  
    }  
  
    app.UseStaticFiles();  
  
    app.UseMvcWithDefaultRoute();  
  
    app.Run(async (context) =>  
    {  
        await context.Response.WriteAsync("Hello World!");  
    });  
}

Notice, we placed UseStaticFiles() middleware before UseMvcWithDefaultRoute()middleware. This order is important, because if the request is for a static file like an image, css or JavaScript file, then UseStaticFiles() middleware will handle the request and short-circuit the rest of the pipeline.    
  
So if the request is for a static file, UseMvcWithDefaultRoute() middleware is not executed, there by avoiding the unnecessary processing.    
  
On the other hand, if the request is an MVC request, UseStaticFiles() middleware will pass that request to UseMvcWithDefaultRoute() middleware which will handle the request and produces the response.   
  
Notice, in addition to UseMvcWithDefaultRoute() middleware, we also have UseMvc()middleware. For now, let's use, UseMvcWithDefaultRoute() middleware. In our upcoming sessions, when we discuss routing, we will discuss the difference between these 2 middlewares.   
  
At this point, if we run the application and navigate to the root URL -http://localhost:49119, we see "Hello World!" message displayed in the browser.

* With UseMvcWithDefaultRoute() middleware configured in the pipeline, when we issue a request to the root URL - http://localhost:49119
* As the request is not for a static file, UseStaticFiles() middleware will pass the request to UseMvcWithDefaultRoute() middleware
* Since we have not specified the controller and action method segments in the URL, UseMvcWithDefaultRoute() middleware looks for Index() action method in the HomeController.
* Since, at the moment, we do not have a HomeController in our application, UseMvcWithDefaultRoute() middleware passes the request to the middleware registered using the Run() method and hence we see the "Hello World!" message produced by this middleware.

Now let's see, what happens if we remove the Run() middleware. At this point the code in your Configure() method must be as shown below.

public void Configure(IApplicationBuilder app, IHostingEnvironment env)  
{  
    if (env.IsDevelopment())  
    {  
        app.UseDeveloperExceptionPage();  
    }  
  
    app.UseStaticFiles();  
  
    app.UseMvcWithDefaultRoute();  
}

With the above code in place, if we again issue a request to the root URL - http://localhost:49119, we now see the 404 error. This is because UseMvcWithDefaultRoute() middleware did not find the HomeController with the Index() action and there is no other middleware in the pipeline, so we see 404 error.  
  
**Add HomeController**  
  
Add Controllers folder, in the root project folder. In the "Controllers" add a new Controller. Copy and paste the following code.

public class HomeController  
{  
    public string Index()  
    {  
        return "Hello from MVC";  
    }  
}

Build the solution and issue a request to the application root URL - http://localhost:49119. You will now see the string - "Hello from MVC" displayed in the browser.