using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Http;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

namespace AspNetCore\_MiddlewareDemo

{

public class Startup

{

// This method gets called by the runtime. Use this method to add services to the container.

// For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940

public void ConfigureServices(IServiceCollection services)

{

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

app.UseRouting();

app.UseEndpoints(endpoints =>

{

endpoints.MapGet("/", async context =>

{

await context.Response.WriteAsync("Hello World!");

});

});

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Http;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

namespace AspNetCore\_MiddlewareDemo

{

public class Startup

{

// This method gets called by the runtime. Use this method to add services to the container.

// For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

**public void Configure(IApplicationBuilder app)**

**{**

**app.Run(async context =>**

**{**

**await context.Response.WriteAsync("hello");**

**});**

**}**

}

}

----------------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Http;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

namespace AspNetCore\_MiddlewareDemo

{

public class Startup

{

// This method gets called by the runtime. Use this method to add services to the container.

// For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

**public void Configure(IApplicationBuilder app)**

**{**

**app.Run(async context =>**

**{**

**await context.Response.WriteAsync("1st hello");**

**});**

**app.Run(async context =>**

**{**

**await context.Response.WriteAsync("2nd hello");**

**});**

**}**

}

}

-----------------------------------------

Using Chainig

**public void Configure(IApplicationBuilder app)**

**{**

**app.Use(async (context, next) =>**

**{**

**await context.Response.WriteAsync("1st hello");**

**await next();**

**});**

**app.Run(async context =>**

**{**

**await context.Response.WriteAsync("2nd hello");**

**});**

**}**

--------------------------------------------

public void Configure(IApplicationBuilder app)

{

**app.Use(async (context, next) =>**

**{**

**await context.Response.WriteAsync("1st hello");**

**await next();**

**await context.Response.WriteAsync("After last");**

**});**

**app.Run(async context =>**

**{**

**await context.Response.WriteAsync("2nd hello");**

**});**

---------------------------------------

public void Configure(IApplicationBuilder app)

{

**app.Map("/newbranch", a => a.Run(c => c.Response.WriteAsync("Running from the newbranch branch!")));**

**app.Use(async (context, next) =>**

**{**

**await context.Response.WriteAsync("Hello, World!");**

**await context.Response.WriteAsync("<br>This is after another Hello, World!");**

**//await next();**

**});**

**app.Run(async context =>**

**{**

**await context.Response.WriteAsync("<br>222222Another Hello, World!");**

**});**

**}**

--------------------------

Routing

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using AspNetCore\_Demo.Models;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.HttpsPolicy;

using Microsoft.EntityFrameworkCore;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

namespace AspNetCore\_Demo

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

services.AddDbContext<EmployeeDbContext>(op => op.UseSqlServer(Configuration["ConnectionStrings:EmployeeDbContext"]));

services.AddControllersWithViews();

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

**app.UseEndpoints(endpoints =>**

**{**

**endpoints.MapControllerRoute(**

**name: "default",**

**pattern: "{controller=Home1}/{action=Index}/{id?}");**

**endpoints.MapControllerRoute(**

**name: "first",**

**pattern: "{controller=Employees}/{action=Index}/{id?}");**

**});**

}

}

}