



Asp.NET Core MVC

ARCTECH INFO PRIVATE LTD

MVC New Project

Create a new project

Recent project templates

Search for templates (Alt+S)



[Clear all](#)

C#



Windows



Web



ASP.NET Core Web App (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

MVC New Project Options

Additional information

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Windows Cloud Service Web

Framework ⓘ

.NET 5.0 (Current) ▾

Authentication type ⓘ

None ▾

☒ Configure for HTTPS ⓘ

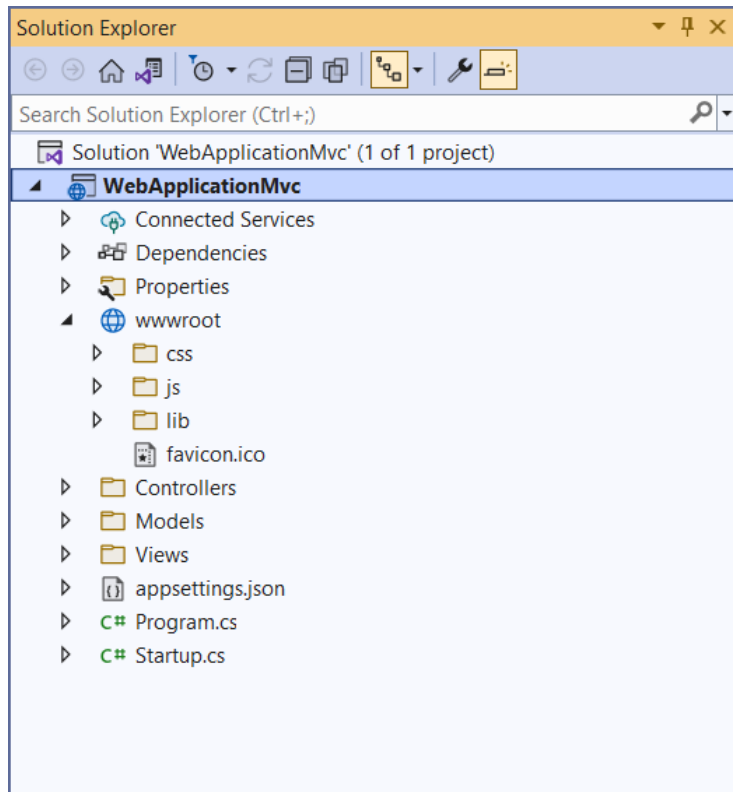
☐ Enable Docker ⓘ

Docker OS ⓘ

Linux ▾

☒ Enable Razor runtime compilation ⓘ

MVC Project Structure



Important project files & folders

- Files
 - appsettings.json
 - Program.cs
 - Startup.cs
- Folders
 - wwwroot
 - Model
 - View
 - Controller

Startup.cs

Startup.cs file is used to manage application server level configurations

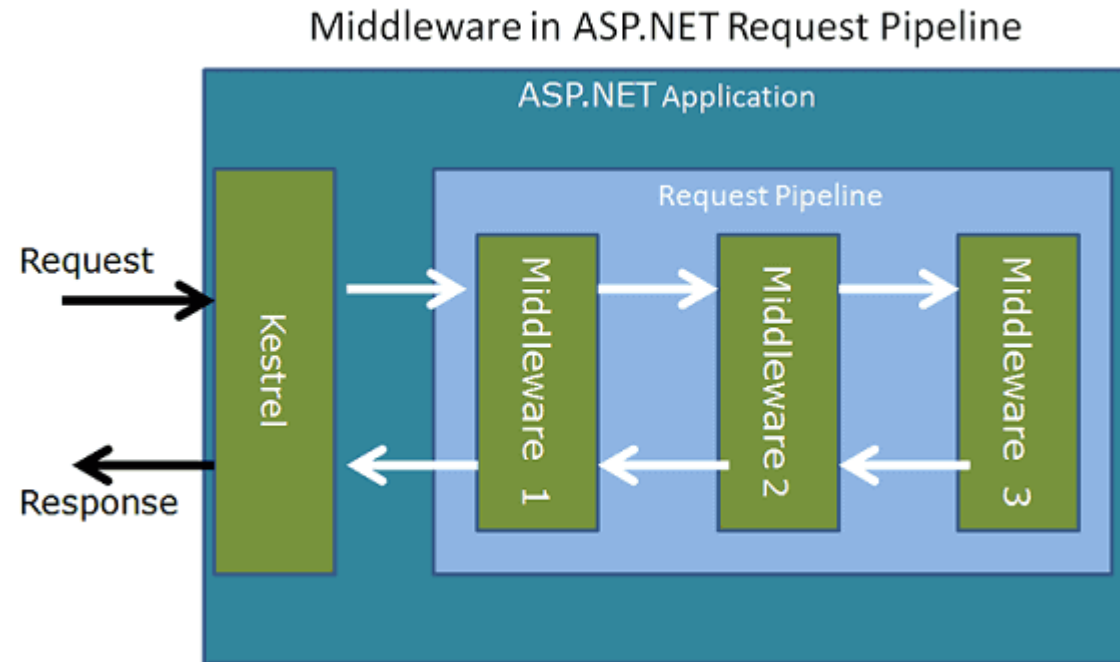
1. Configure method

- Configure the HTTP request pipeline

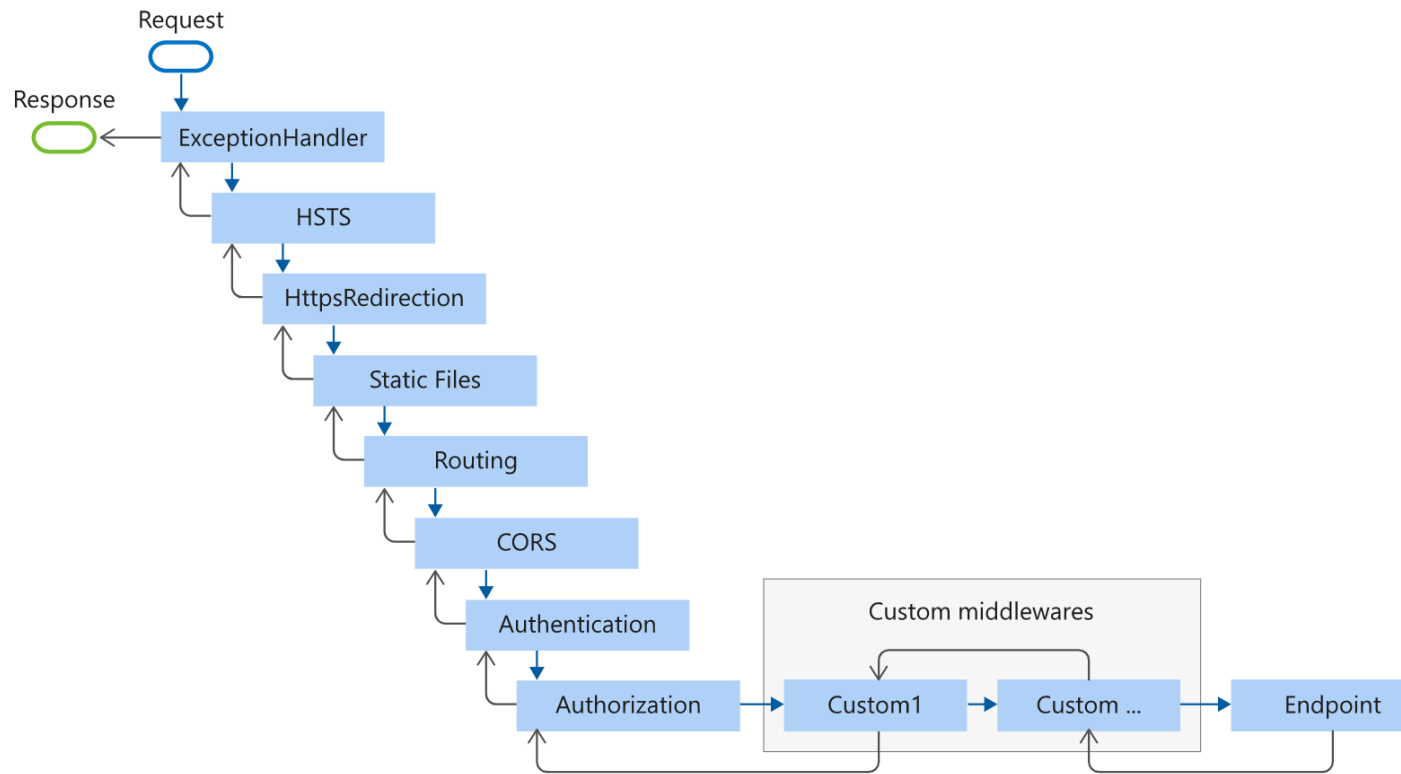
2. ConfigureServices method

- Add services to the application server container

.NET Core Pipeline



.NET Core Pipeline



Adding Middleware to Pipeline

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline
0 references
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 scenario
        app.UseHsts();
    }
    app.UseHttpsRedirection();
    app.UseStaticFiles();

    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}");
    });
}
```

Middleware are code modules which are executed for every http request

The middleware is executed in the same sequence as it appears in the Configure method.

Visual Studio automatically inserts standard middleware modules to a new .NET Core project

Middleware Modules can

- Access & Modify the request parameters
- Return immediately e.g. trying to access Account Balance page without logging in.
- Access & Modify the response data

You can create your own Custom Middleware module and insert it into the pipeline

Asp.NET Framework vs .NET 5

.NET FRAMEWORK ASPX

Views are files with extension .aspx

UserControls => .ascx files

All static and dynamic pages (aspx) are in the same folders

When user types <http://domain-name:port/Home/Index.aspx>, ASP.NET will

- look for a Index.aspx file inside the Home folder
- execute the server side code and server controls
- Generate the output html and return to browser

.NET 5 RAZOR VIEW

Views are files with extension .cshtml

Partial Views => _[ViewName]Partial.cshtml

Static files are in wwwroot folder, while dynamic views (cshtml) are outside

When user types <http://domain-name:port/Home/Index>, MCV .NET 5 will

- Look for a controller class called Home
- Execute the action method called Index
- The return value of the action method is packaged into an html and returned to the browser

Routing Overview

In MVC, routing is a process of mapping the browser request to the controller action and return response back.

Each MVC application has default routing for the default HomeController.

Routing is handled in MVC applications via the `app.MapControllerRoute` middleware

```
◦ app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}");
});
```

The important section here is the controller, action and id parameter

More on routing

A general URL in asp.net looks like <http://domain-name:port/{folder-name}/{page-name}?id=5>

- For e.g., <http://localhost:5555/customer/ViewCustomer.aspx?id=5>

In MVC however the URL is <http://domain-name:port/{controller}/{action}/{id}>

- For e.g., <http://localhost:5555/customer/showdata/5>
- This will execute the ShowData method inside the CustomerController class and return the desired View.
- In the example below, return View() returns the default view which is ShowData to the browser

```
public class CustomerController : Controller
{
    public IActionResult ShowData()
    {
        return View();
    }
}
```