**Middle ware**

Middle ware ها در واقع یک سری component هستند که در مسیر دریافت request و ارسال response قرار می گیرند و می توانند روی آن ها تغییراتی ایجاد کنند.

برای درک بهتر نحوه کارکرد middle ware ها ، می توان عملکرد آن را به یک سیستم تصفیه آب تشبیه کرد. این سیستم شامل چندین مخزن است، آب وارد این سیستم می شود در هر مخزن ممکن است چیزهایی از آن گرفته شود یا به آن اضافه شود و سپس وارد مخزن بعدی شود و در نهایت آب تصفیه شده از سیستم خارج شود.

****

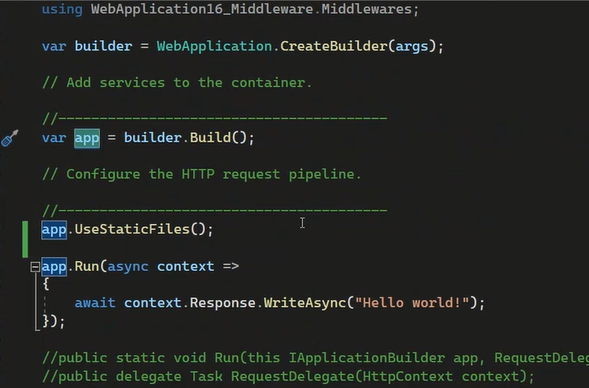
در .Net ، در نسخه های قبل از Core قابلیت Middle ware وجود نداشت. در نسخه های core در ورژن های قبل از 6 در فایل startup و در 6 به بعد در فایل program.cs ، اقدام به معرفی middle ware می کنیم.

یک مثال ساده از middle weare : پوشه wwwroot که در آن فایل های static مثل html,css,image,… را قرار می دهیم و برنامه آن ها را در روت درنظر می گیرد به شرطی که Middle ware آن را در کانفیگ برنامه با دستور زیر اضافه کنیم:

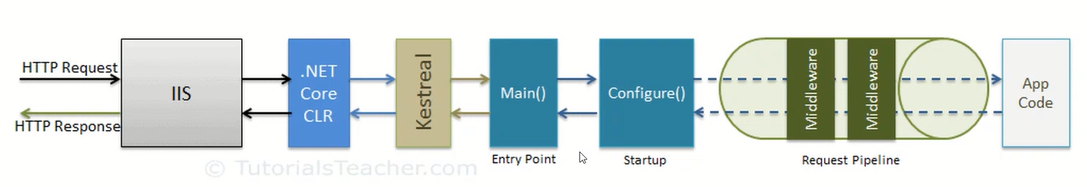
app.UseStaticFiles();

**محل قرارگیری دستورات middleware**

طبق تصویر زیر؛ ابتدا builder تعریف شده و Service ها به آن add می شود. سپس دستور app = builder.Build(); را داریم و در ادامه Middleware ها را به app اضافه می کنیم. (ترتیب این دو عمل مهم است)



**تصویری از جایگاه Middle ware در برنامه**



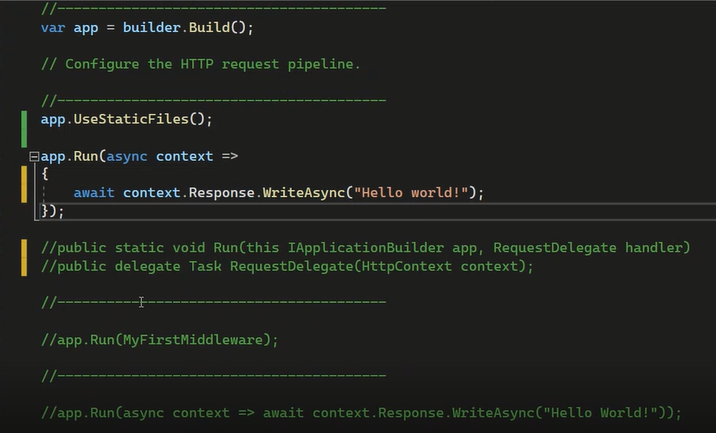
**انواع**

برای استفاده از middleware سه Extention Method به نام های app.Run ، app.Use و app.Map داریم. در ادامه به توضیح این متدها و تفاوت هایشان با هم می پردازیم:

**app.Run**

اصطلاحا به آن Terminal Middle Ware می گویند. یعنی در واقع باید آخرین Middle ware باشد چون به محض اینکه درخواست به این Middle ware می رسد خاتمه پیدا می کند و دیگر به Middle ware های بعدی نمی رسد. در اینجا از اصطلاح short circuit (اتصال مدار کوتاه) استفاده می کنیم.

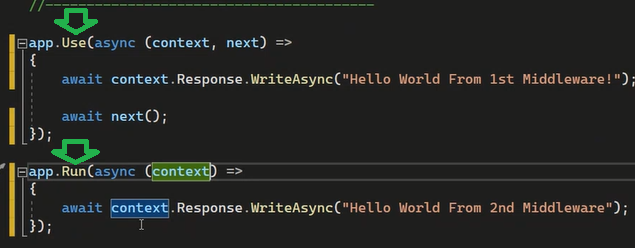
مثال: در این مثال، اگر ما دستورات دیگری بعد از app.Run بنویسیم، برنامه به آنها نمی رسد و اجرا نمی شوند. چون آخرین دستوری که اجرا می شود app.Run است. وقتی برنامه را اجرا می کنیم آدرس هر صفحه ای را که بزنیم Hello world را نمایش می دهد.



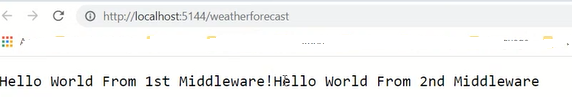
**app.Use**

این Middle ware یک حالت معمولی است. برخلاف نوع قبلی که خاتمه دهنده بود، Next دارد و در ادامه آن، برنامه Middle ware های بعدی را فراخوانی می کند.

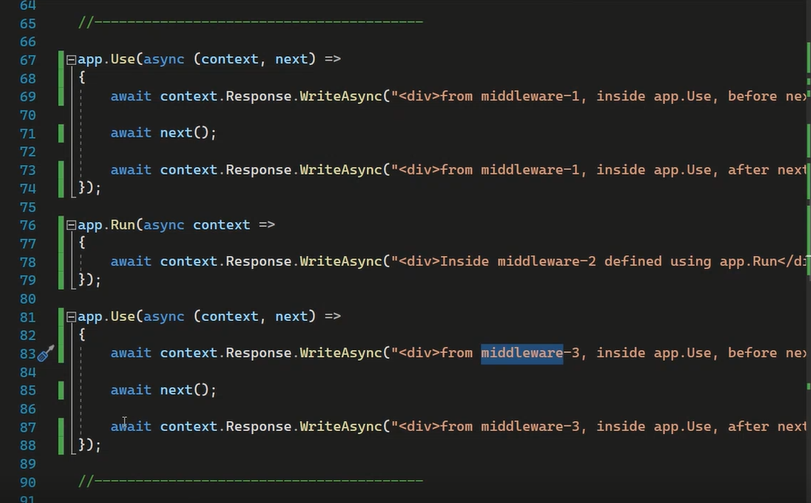
🡄 اگر دوتا app.Run داشته باشیم اولی اجرا می شود و به دومی نمی رسد. پس برای اجرای هر دو Middle ware باید اولی را app.Use و دومی را app.Run بنویسیم.



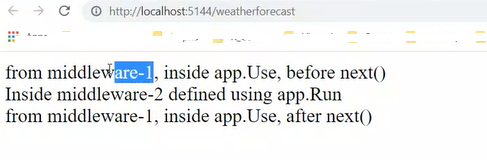
خروجی:



یک مثال:



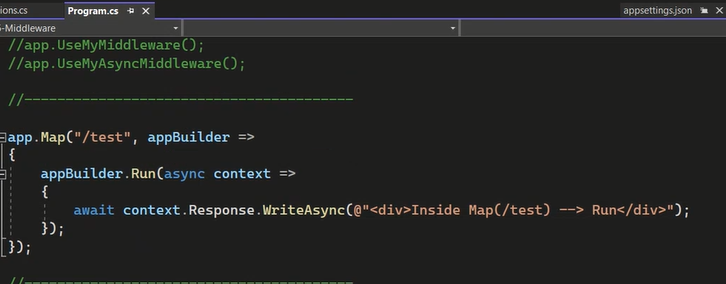
خروجی:



**app.Map**

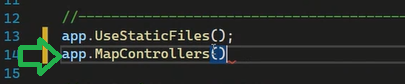
اصطلاحا به آن branch می گویند. براساس درخواستی که آمده و اینکه به چه مسیری درخواست داده، اگر در آن مسیر بود (یعنی در اینجا یک شرط چک می شود) آن Middle ware را اجرا می کند و اگر نبود اجرا نمی کند.

در نمونه زیر، وقتی برنامه را اجرا می کنیم خطا می دهد، ولی اگر در ادامه url ، عبارت /test را بنویسیم، نتیجه را برمی گرداند:





نکته ) مثلا برای استفاده از کنترلرها از این نوع Middle ware استفاده می کنیم:

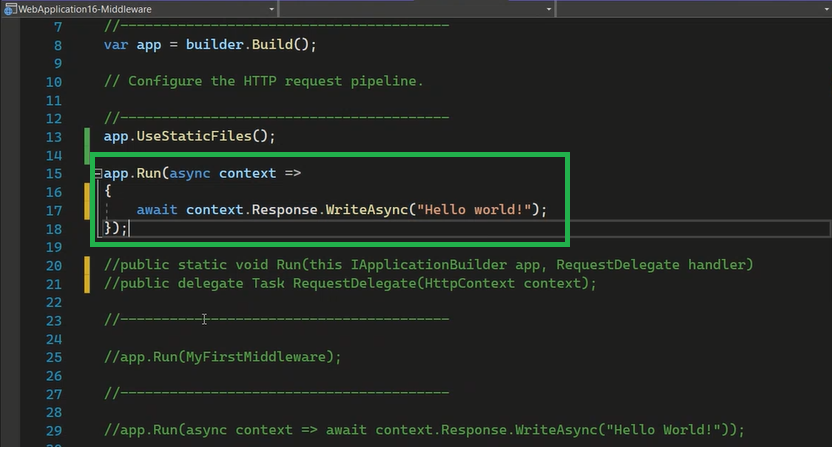


**برای ساخت Middleware**

دو راه وجود دارد: 1- inline 2- غیر inline

**نحوه نوشتن به صورت inline**

مثال: به صورت **Anonymous نوشته می شود. مانند تصویر زیر که در program نوشته و فراخوانی شده است.**



**نحوه نوشتن به صورت غیر inline**

**یک متد مجزا می نویسیم و سپس فراخوانی می کنیم:**

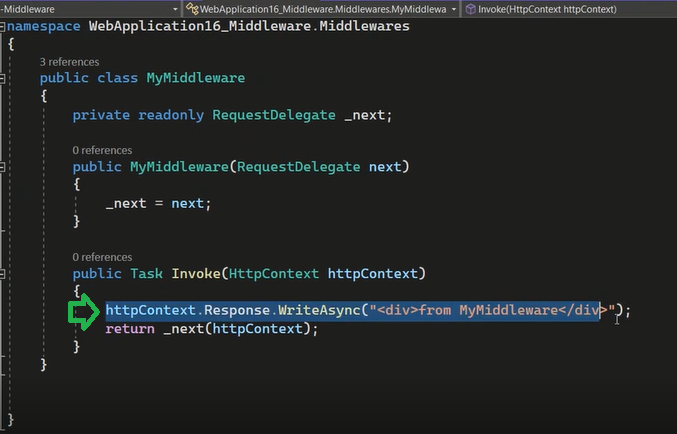


🡄 اگر بخواهیم به روش فوق، Middle ware ها را در program.cs بنویسیم این فایل شلوغ می شود و کد ما تمیز نیست. پس Middle ware را در یک کلاس جدا می نویسیم و در program.cs به ترتیبی که می خواهیم Middle ware ها اجرا شوند، آنها را فراخوانی می کنیم.

**نحوه نوشتن Middleware خارج از program.cs**

Middleware به صورت یک class نوشته می شود. کلاسی که در سازنده آن یک پارامتر next از نوع **RequestDelegate** در نظر می گیریم، برای اینکه بتوانیم با استفاده از آن Middleware بعدی را صدا بزنیم. هم چنین باید یک متد **Invoke** داشته باشد که **HttpContext** را به عنوان ورودی بگیرد.

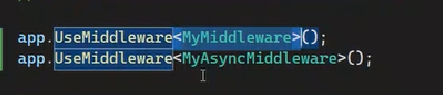
در قسمت مشخص شده، یعنی قبل از next کدهایمان را می نویسیم:



هم چنین می توان متد فوق را به صورت async نوشت.

\* اگر بخواهیم یک اتصال مدار کوتاه ایجاد کنیم، در قطعه کد بالا به جای \_next باید Task.CompleteTask را Return کنیم.

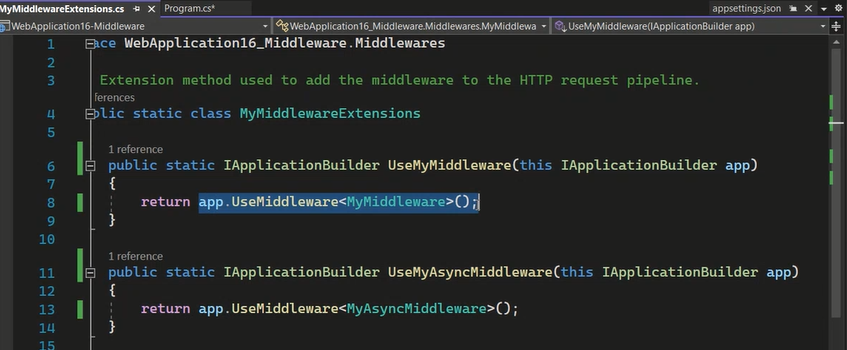
🡄 برای معرفی Middleware ای که به صورت class مجزا نوشته شده به pipeline ، در program.cs از دستور **UseMiddleware** که برای خود framework است، استفاده می کنیم و اسم Middleware را به صورت Generic به آن می دهیم:



این امکان که چندین بار middleware را با UseMiddleware فراخوانی کنیم وجود دارد.

نکته) خود فریم ورک .net از دستور UseMiddleWare استفاده نمی کند. بلکه برای MiddleWare هایش یک Extesion می نویسد و آن را فراخوانی می کند. مثلا app.UseStaticfile..

در ادامه یک نمونه می نویسیم.



در قطعه کد بالا یک کلاس static با یک extension method نوشتیم. (داخل پرانتز متدی که در ورودی this می گیرد یک Extension Method است) ورودی و خروجی این متد از جنس IApplicationBuilder است و دستور UseMiddleWare را به داخل این متد منتقل می کنیم. در program با دستور زیر آن را فراخوانی می کنیم:

app.Use**MyMiddleware**();

به Extenstion متد تعریف شده می توان option اضافه کرد و در هنگام فراخوانی پارامتر پاس کرد. (در ویدئوهای 7 و 8 حکیم قدس توضیح داده شده است.)

\* استفاده از middleware منجر به بهینگی کد می شود.

برای درک بهتر مفهوم Middle ware ویدئوهای حکیم قدس در یوتیوب را مشاهده کنید.

<https://www.youtube.com/watch?v=hp8DnFUarAI&list=PLbtjX995c0WrcX4g_rCJTqoitZpLRLhDx&ab_channel=%DA%A9%D8%AF%D9%87%DA%A9>