



Asp.NET Core Middleware



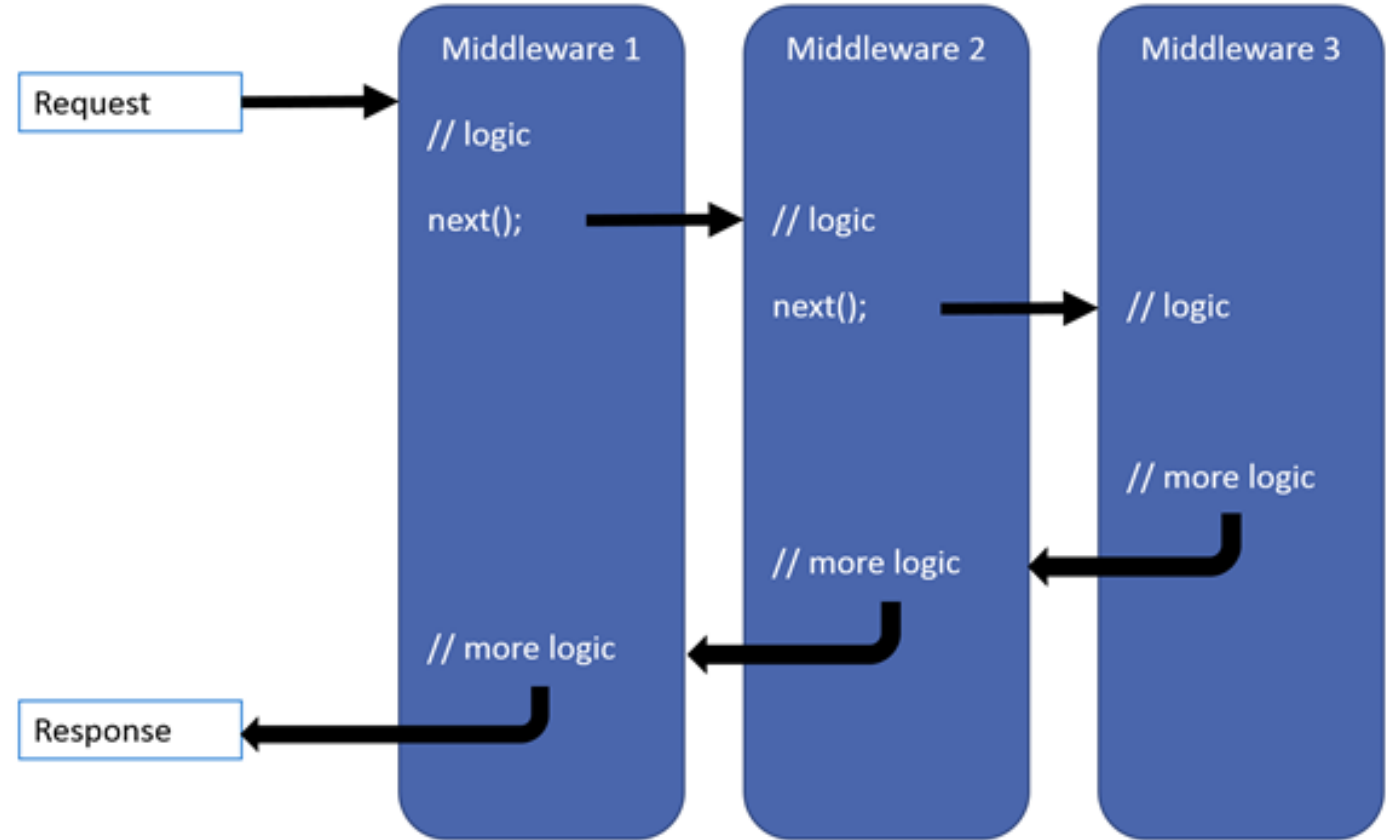
Asp.NET Core Middleware Nedir?

- Web uygulamasına client'tan gelen request'e karşılık verilecek response'a kadar arada farklı işlemler gerçekleştirmek ve sürecin gidişatına farklı yön vermek isteyebiliriz.



Middleware'ler Nasıl Çalışır?

- Middleware'ler sarmal bir şekilde tetiklenirler.



Asp.NET Core Uygulamasında Middleware

Asp.NET Core, yapısal olarak middleware yapılanmasını destekleyen bir çekirdeğe sahiptir.

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseSwagger();
        app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.json", "MiddlewareExample v1"));
    }

    app.UseHttpsRedirection();

    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllers();
    });
}
```



Hazır Middleware'ler

- Run
- Use
- Map
- MapWhen

Run Metodu

- Run fonksiyonu kendisinden sonra gelen middleware'ı tetiklemez!
- Dolayısıyla kullanıldığı yerden sonraki middleware'ler tetiklenmeyeceğinden dolayı akış kesilecektir.
- Bu etkiye Short Circuit(Kısa Devre) denir.

```
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseSwagger();
        app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.json", "Middleware 1"));
    }

    app.Run(async c => { await c.Response.WriteAsync("Middleware 1"); });
    app.Run(async c => { await c.Response.WriteAsync("Middleware 2"); });

    app.UseHttpsRedirection();

    app.UseRouting();

    app.UseAuthentication();
}
```

localhost:5001/weatherforecast

Middleware 1

Use Metodu

- Run metoduna nazaran, devreye girdikten sonra süreçte sıradaki middleware'i çağırmakta ve normal middleware işlevi bittikten sonra geriye dönüp devam edebilen bir yapıya sahiptir.

```
app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.j  
}  
  
app.Use(async (context, task) =>  
{  
    Console.WriteLine("Start Use Middleware");  
    await task.Invoke();  
    Console.WriteLine("Stop Use Middleware");  
});  
app.Run(async c =>  
{  
    Console.WriteLine("Start Run Middleware");  
});  
  
app.UseHttpsRedirection();
```

```
Start Use Middleware  
Start Run Middleware  
Stop Use Middleware
```

```
app.Use(async (context, task) =>
{
    Console.WriteLine("Start Use Middleware 1");
    await task.Invoke();
    Console.WriteLine("Stop Use Middleware 1");
});
app.Use(async (context, task) =>
{
    Console.WriteLine("Start Use Middleware 2");
    await task.Invoke();
    Console.WriteLine("Stop Use Middleware 2");
});
app.Use(async (context, task) =>
{
    Console.WriteLine("Start Use Middleware 3");
    await task.Invoke();
    Console.WriteLine("Stop Use Middleware 3");
});
```

```
Start Use Middleware 1
Start Use Middleware 2
Start Use Middleware 3
Stop Use Middleware 3
Stop Use Middleware 2
Stop Use Middleware 1
```


Map Metodu

- Bazen middleware'i talep gönderen path'e göre filtrelemek isteyebiliriz. Bunun için Use ya da Run fonksiyonlarında if kontrolü sağlayabilir yahut Map metodu ile daha profesyonel işlem yapabiliriz.

```
app.Use(async (context, task) =>
{
    Console.WriteLine("Start Use Middleware 1");
    await task.Invoke();
    Console.WriteLine("Stop Use Middleware 1");
});

app.Map("/weatherforecast", builder =>
{
    builder.Run(async c => await c.Response.WriteAsync("Run middleware'i tetiklendi"));
});

app.Map("/home", builder =>
{
    builder.Use(async (context, task) =>
    {
        Console.WriteLine("Start Use Middleware2 ");
        await task.Invoke();
        Console.WriteLine("Stop Use Middleware2 ");
    });
});
```

MapWhen Metodu

- Map metodu ile sadece request'in yapıldığı path'e göre filtreleme yapılırken, MapWhen metodu ile gelen request'in herhangi bir özelliğine göre bir filtreleme işlemi gerçekleştirilebilir.

```
app.MapWhen(c => c.Request.Method == "GET", builder =>
{
    builder.Use(async (context, task) =>
    {
        Console.WriteLine("Start Use Middleware2 ");
        await task.Invoke();
        Console.WriteLine("Stop Use Middleware2 ");
    });
});
```


Özel Middleware Oluşturma

