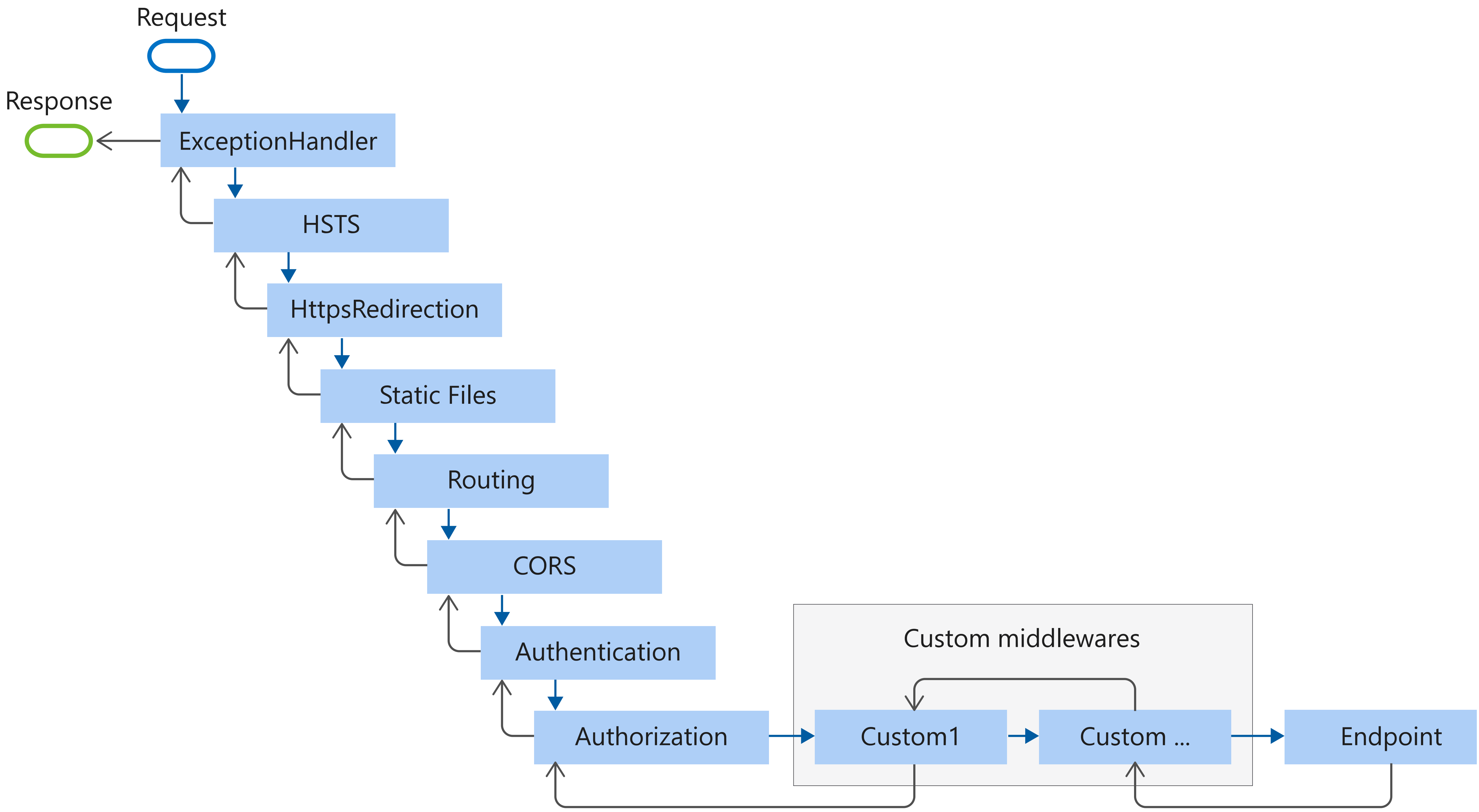
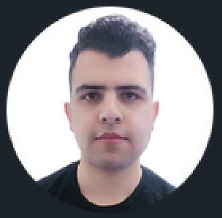
ASP.NET Middleware Ordering

In ASP.NET Core, middleware executes in the order it's registered. Incorrect ordering can break authentication, security, and routing.



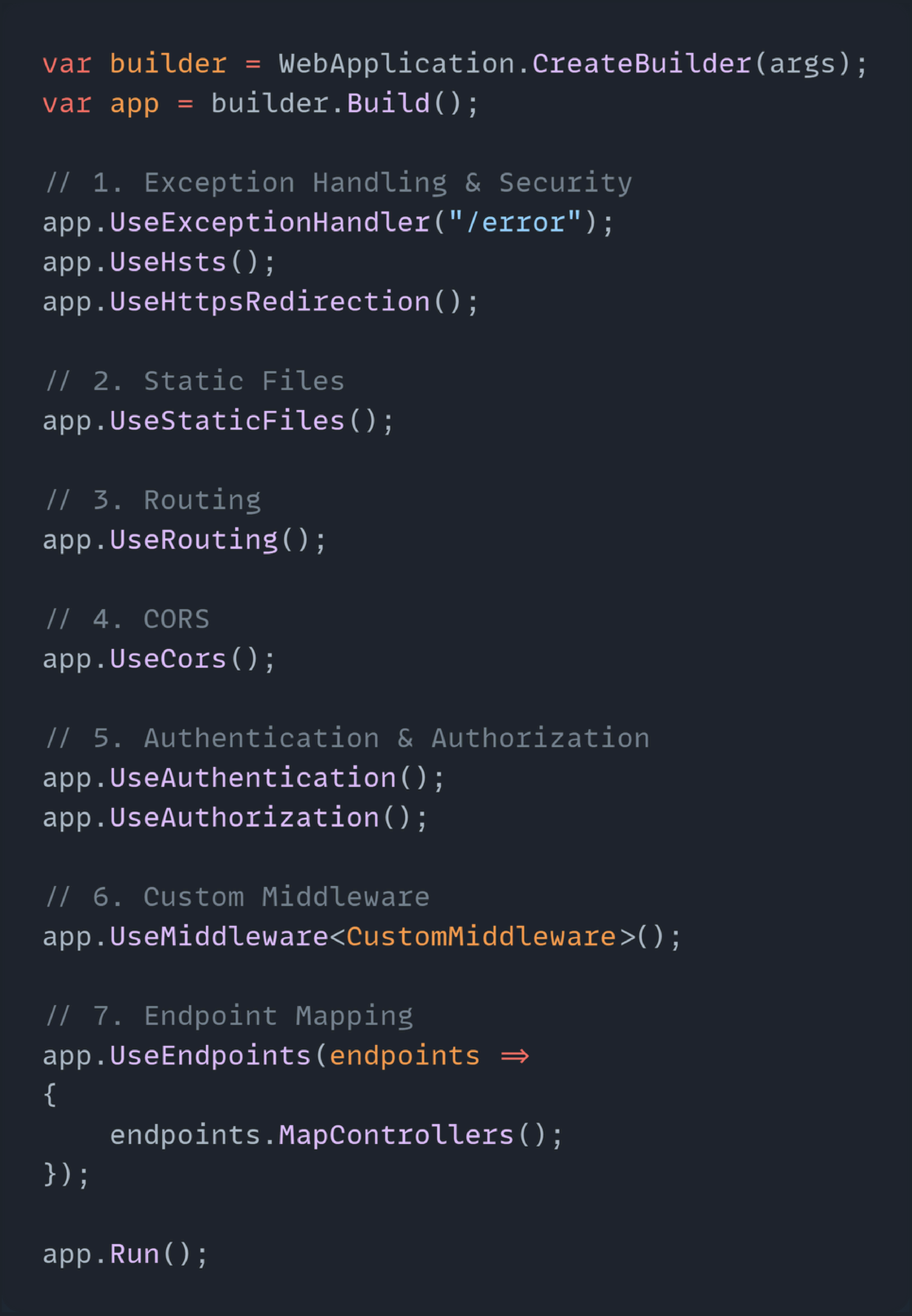
[src: https://learn.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-9.0]



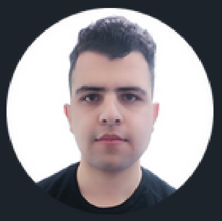
Elliot One

Middleware Ordering Sample

One of the biggest mistakes I see developers make is misordering middleware.



```csharp
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

// 1. Exception Handling & Security
app.UseExceptionHandler("/error");
app.UseHsts();
app.UseHttpsRedirection();

// 2. Static Files
app.UseStaticFiles();

// 3. Routing
app.UseRouting();

// 4. CORS
app.UseCors();

// 5. Authentication & Authorization
app.UseAuthentication();
app.UseAuthorization();

// 6. Custom Middleware
app.UseMiddleware<CustomMiddleware>();

// 7. Endpoint Mapping
app.UseEndpoints(endpoints =>
{
    endpoints.MapControllers();
});

app.Run();
```



Elliot One

Elliot One

Enjoyed Reading This?

Reshare and Spread Knowledge.