

# Middleware in .NET (Beginner Guide)

In ASP.NET Core, a Middleware is a software component that is assembled into the application pipeline to handle requests and responses. Each middleware has the option to either process the request itself or pass it to the next middleware in the pipeline.

## ■ Where Middleware is Used?

- Authentication & Authorization (e.g., checking if the user is logged in).
- Logging and Monitoring (e.g., recording request/response details).
- Error Handling (e.g., showing custom error pages).
- Request Routing (e.g., directing requests to the correct controller/action).
- CORS, Caching, Compression and more.

## ■■ How to Create a Middleware in .NET?

- 1 Step 1: Create a class with a constructor that accepts RequestDelegate.
- 2 Step 2: Implement an Invoke or InvokeAsync method that processes HttpContext.
- 3 Step 3: Register the middleware in Program.cs using app.UseMiddleware().

## ■ Example Middleware:

```
public class CustomMiddleware
{
    private readonly RequestDelegate _next;

    public CustomMiddleware(RequestDelegate next)
    {
        _next = next;
    }

    public async Task InvokeAsync(HttpContext context)
    {
        // Do something before request
        Console.WriteLine("Request handled: " + context.Request.Path);

        await _next(context); // Call the next middleware

        // Do something after response
        Console.WriteLine("Response sent: " + context.Response.StatusCode);
    }
}

// Register in Program.cs
app.UseMiddleware<CustomMiddleware>();
```