.NET TIP

3 Ways to Create Custom Middleware in ASP.NET Core

POORNA SOYSA

SWIPE

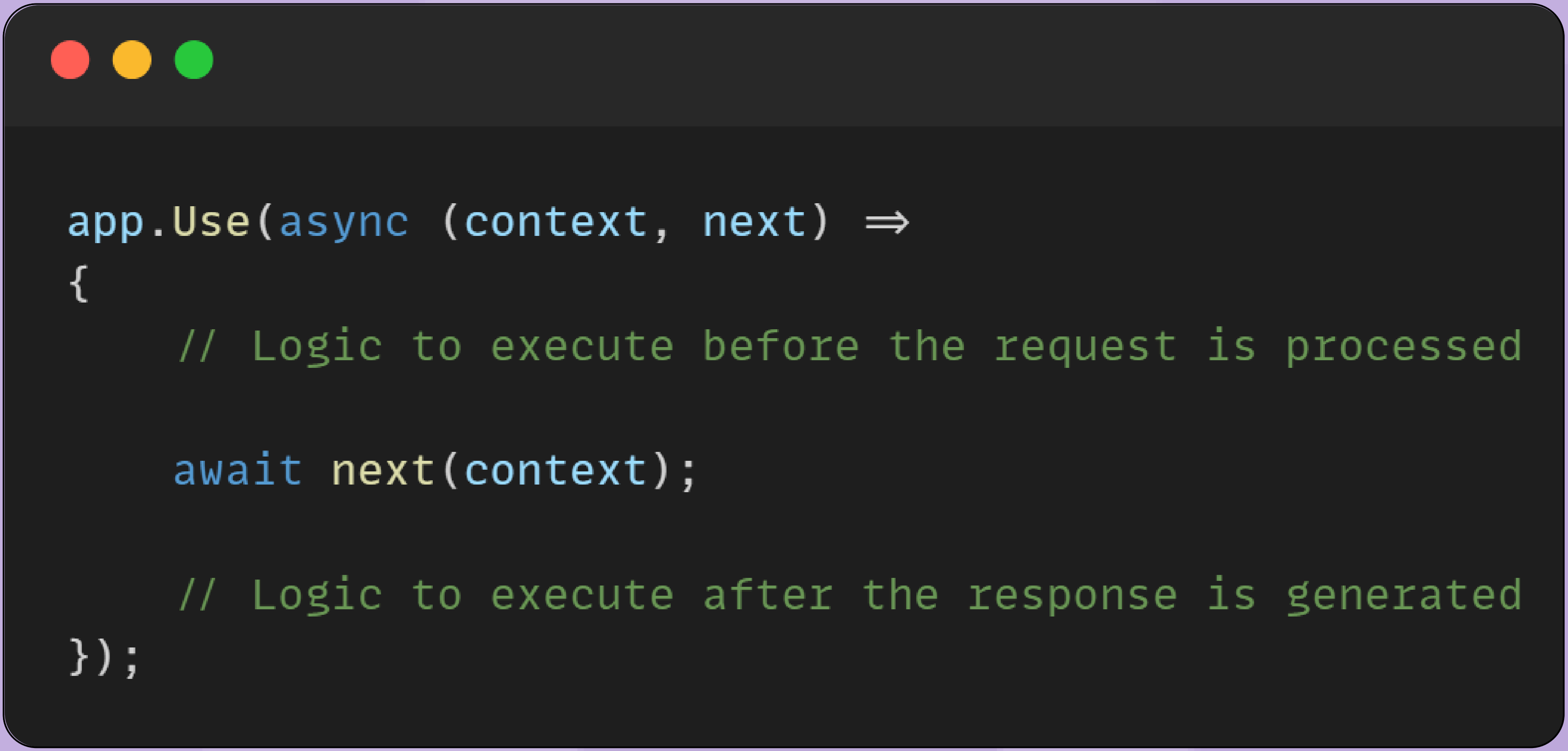
Using Request Delegate

Middleware can be created with a **request delegate** that takes an **HttpContext** as input and returns a **Task**. This method is commonly employed for shorter middleware components.

POORNA SOYSA

SWIPE

Using Request Delegate



```
app.Use(async (context, next) ⇒
{
    // Logic to execute before the request is processed

    await next(context);

    // Logic to execute after the response is generated
});
```

POORNA SOYSA

SWIPE

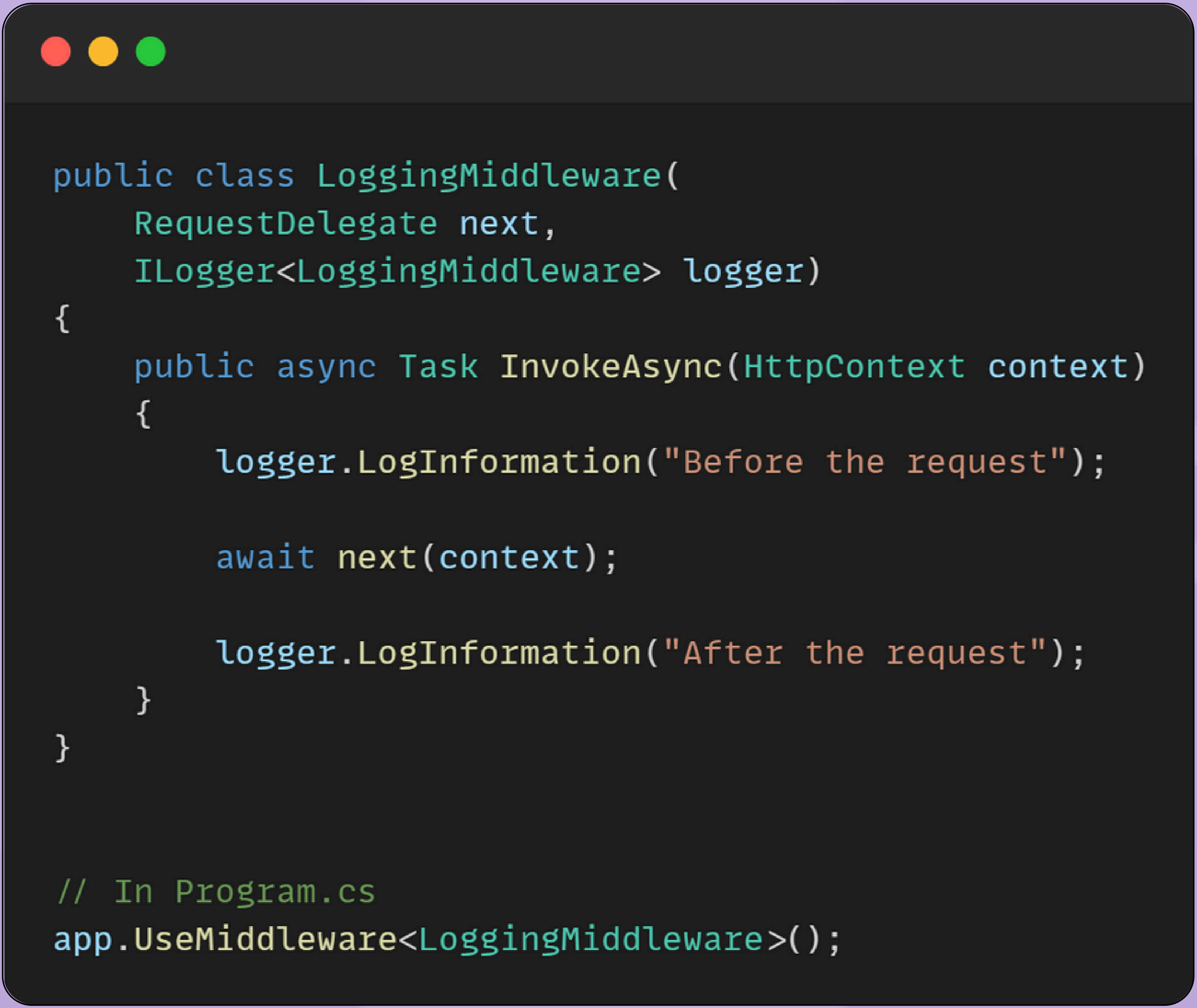
Convention-based Middleware

A middleware can be extracted to a separate class that adheres to the specific convention.

POORNA SOYSA

SWIPE

Convention-based Middleware



```
public class LoggingMiddleware(
    RequestDelegate next,
    ILogger<LoggingMiddleware> logger)
{
    public async Task InvokeAsync(HttpContext context)
    {
        logger.LogInformation("Before the request");

        await next(context);

        logger.LogInformation("After the request");
    }
}


// In Program.cs
app.UseMiddleware<LoggingMiddleware>();
```

POORNA SOYSA

SWIPE

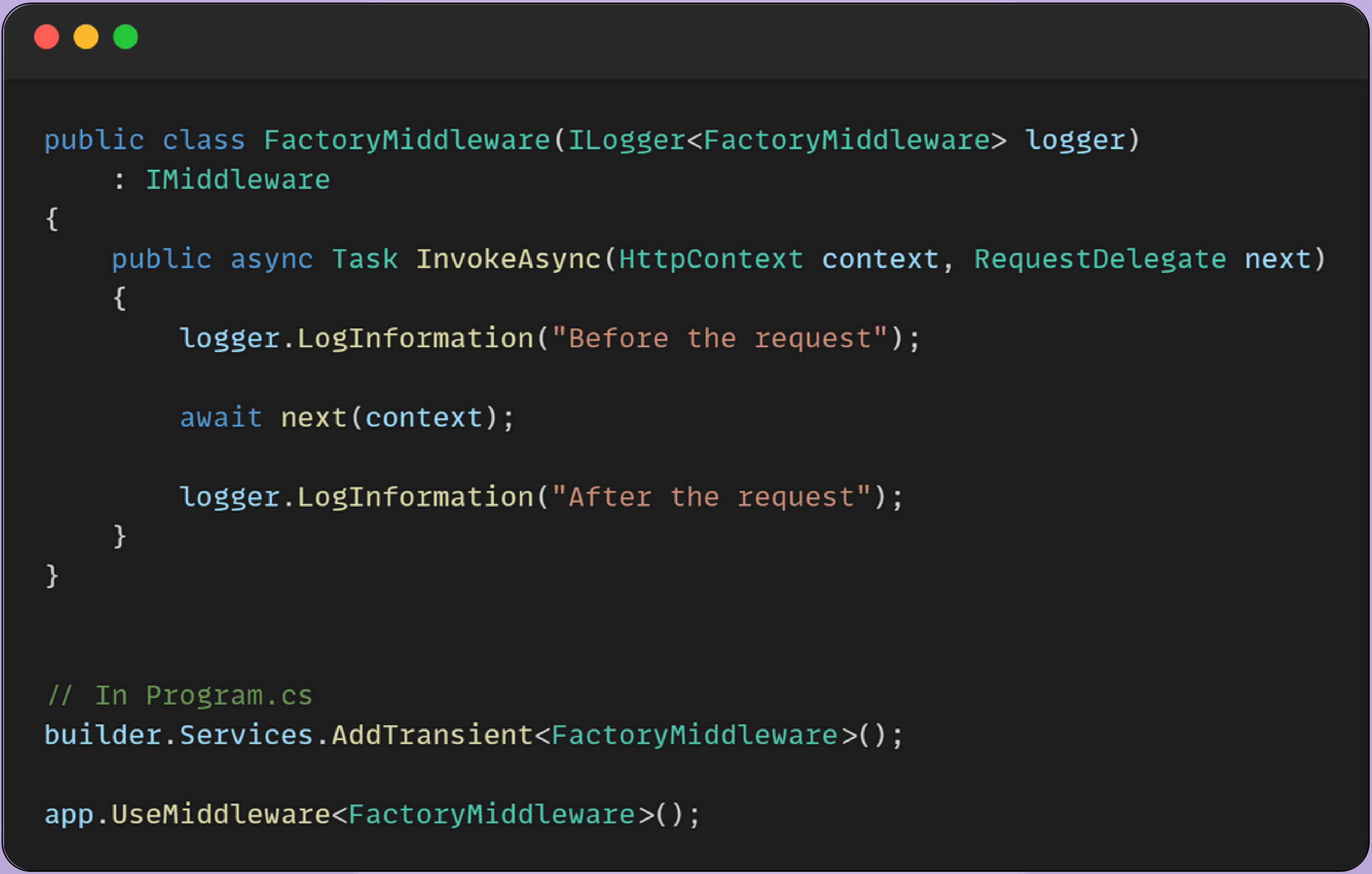
Factory-based Middleware

Create a custom middleware component that implements the **IMiddleware** interface.

POORNA SOYSA

SWIPE

Factory-based Middleware



```
public class FactoryMiddleware(ILogger<FactoryMiddleware> logger)
    : IMiddleware
{
    public async Task InvokeAsync(HttpContext context, RequestDelegate next)
    {
        logger.LogInformation("Before the request");

        await next(context);

        logger.LogInformation("After the request");
    }
}


// In Program.cs
builder.Services.AddTransient<FactoryMiddleware>();

app.UseMiddleware<FactoryMiddleware>();
```

POORNA SOYSA

SWIPE

Let's spread the **knowledge** together!

DO YOU LIKE THIS POST?

REPOST IT!

POORNA SOYSA

LEAVE A COMMENT BELOW