

ASP.NET CORE

PIPELINE

issntt@nus.edu.sg

Objectives

At the end of this lesson, students will be able to

- Describe some of cross-cutting concerns of web apps
- Describe the actions that a middleware can do for an HTTP Request or HTTP Response
- Analyze how a pipeline, set up with some common middleware, handles a particular request



When developing web apps, we often need to:

- **Log the requests** from clients and/or **the responses** from our server
- **Distinguish** between requests for **static** files (to **respond right away**) or **dynamic** pages (to **forward to MVC** for further process)
- Handle and respond **error pages** to users

They are called “**cross-cutting concerns**”, because required by **many parts** of the program

How should we implement these kinds of features?

Middleware

In .NET Core, middleware is a **C# class** that can **handle HTTP Requests or Responses**. It can:

For an
Incoming HTTP request

1. **Generate response** for it **right away**, or
2. **Modify it** and/or **pass it on** to another middleware

For an
Outgoing HTTP response

1. **Modify** it, and/or
2. **Pass it on** to another middleware, or to ASP.NET Core web server to response to users

Pipeline

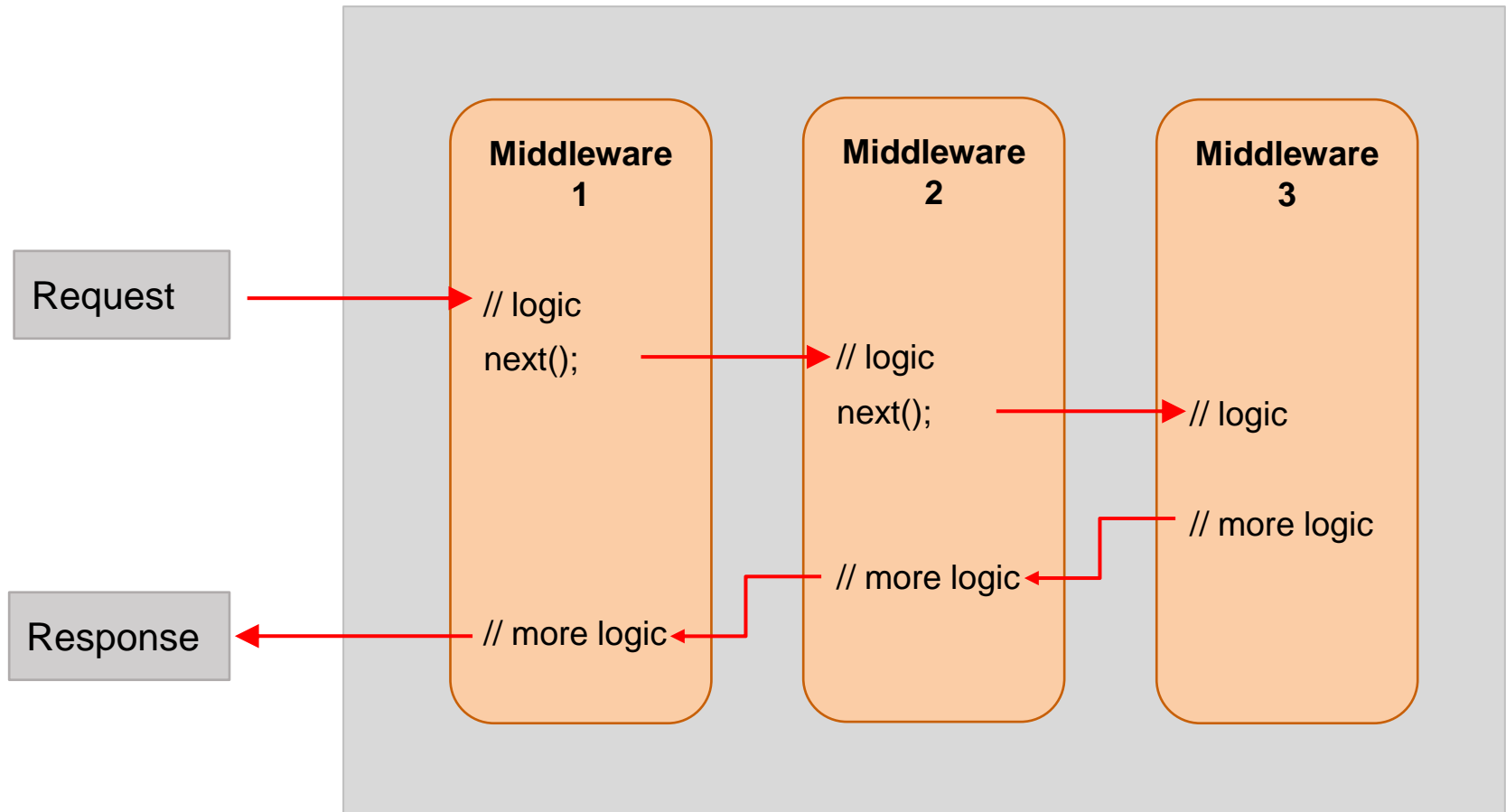
To build a complete application, we **compose multiple middleware** to form a pipeline. Each handles **one aspect of a request** only



Photo by JJ Ying on [Unsplash](#)

Pipeline

Pipeline is **bidirectional** and the **order** of middleware in the pipeline is **important**



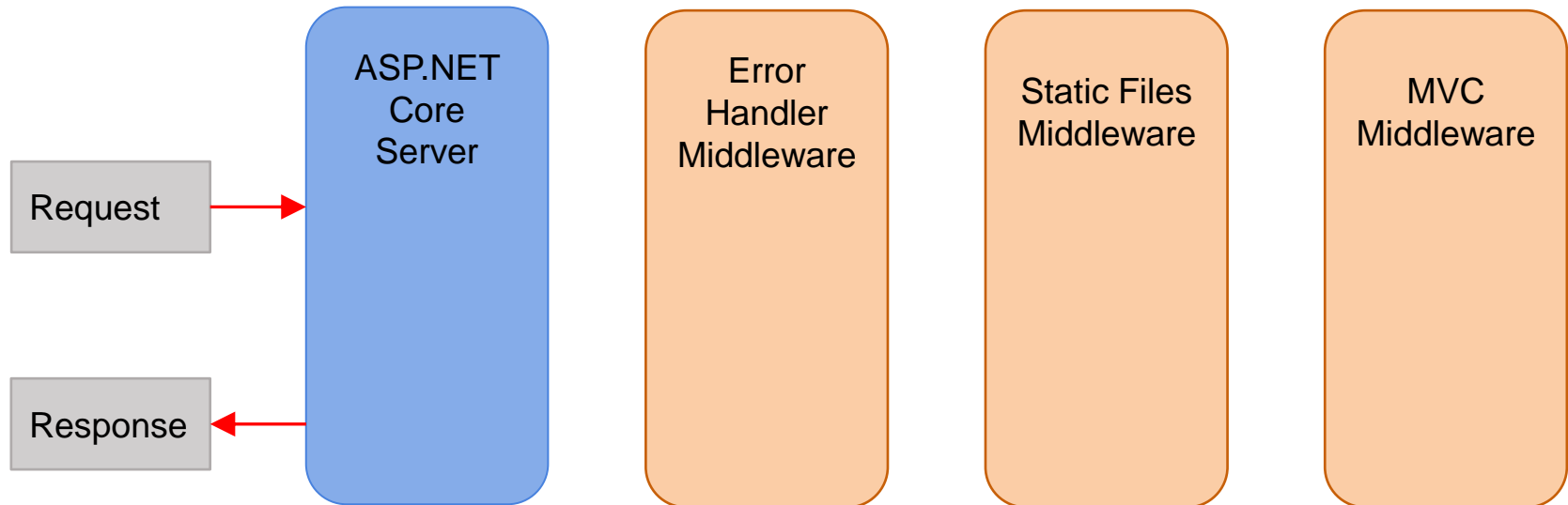
Some common Middleware

- **StaticFileMiddleware**: serve static files from *wwwroot* folder when requested
- **ExceptionHandlerMiddleware**: presents users with a custom error page
- **MVCMiddleware**: processes requests with MVC

<https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-3.1#built-in-middleware>

Pipeline Examples

Let's say we've set up the following pipeline

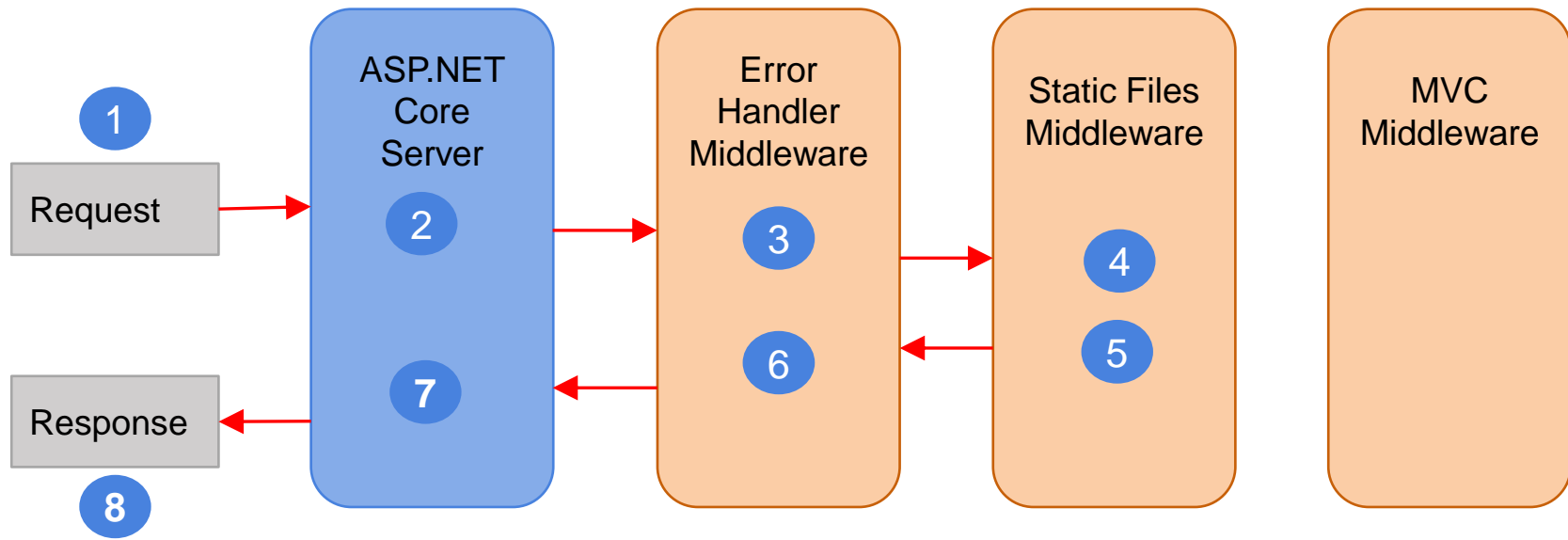


What happens if the request is:

1. for an **existed static file**. E.g. an image?
2. for a **dynamic page** that can be handled by MVC?
3. for something that **cannot be found**?

Pipeline Examples

For an existed static file, *example.js*?



1. Client makes HTTP request for *example.js*

2. ASP.NET Core server handles the request, builds an HttpContext object, and pass it to the Error Handler middleware

3. Error Handler passes to Static Files Middleware

4. The Static Files middleware verifies that the request is for static file and the file existed in **wwwroot/**, so it retrieve the *example.js* and build the response

5. It then **short-circuit** and return the response **without forwarding** to MVC middleware

6. Error Handler Middleware pass the response to the Server

7. ASP.NET Core Server returns the HTTP response to client

8. The HTTP Response containing content of *example.js* is sent to client



Declare Pipeline in ASP.NET Core projects

All middleware **included in the pipeline** will be declared in *Program.cs*. For example

```
...  
app.UseExceptionHandler("/Home/Error");  
  
app.UseStaticFiles();  
  
app.UseRouting();  
  
app.MapControllerRoute(  
    name: "default",  
    pattern: "{controller=Home}/{action=Index}/{id?}");  
...
```



What middleware is being used in this app?

- ASP.NET Core in Action, Chapter 3, Handling requests with the middleware pipeline, *by Andrew Lock*
- What is Middleware?
<https://freecontent.manning.com/what-is-middleware/>
- Middleware Order <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-6.0#middleware-order>