

ASP.NET Core Fundamentals

Manoj Bhargavan

The image shows a complex industrial facility with numerous large, white-painted pipes and valves. The pipes are arranged in a dense, interconnected network, with some running horizontally and others curving upwards or downwards. Several red-handled valves are visible, providing points of control in the system. The background shows more industrial structures under a clear sky. The text "Session 1" is centered over the image in a white, serif font, with a thin white horizontal line positioned directly below it.

Session 1

\$what is dotnet_core

Core



How Does The Web Work?

How Http Request's Work?

Client Initiates A Request and setups up the connection with server

- DNS Lookup
- TCP Handshake
- TLS Handshake / Key exchange

Client sends the HTTP request

- ASCII text
- Details in Request Header Screen Snap

Server Responds based on how is setup

- Response Code and appropriate content
- Client does the job of processing the response, in case of browser it read the response and renders as required, if additional files like images, stylesheets are required the browser request the same in separate request

All this work is abstracted from you as user/developer

▼ Request Headers (578 B)

```
GET /App/About HTTP/1.1
Host: localhost:5000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://localhost:5000/
Cookie: .AspNetCore.Antiforgery.6UcL0rJu9JQ=CfDJ80dmeLcH1ZPnM8x4X9uF-uJDVRESgVd6b2eX0MwCAMu7s6JCoddJXxI
Upgrade-Insecure-Requests: 1
```

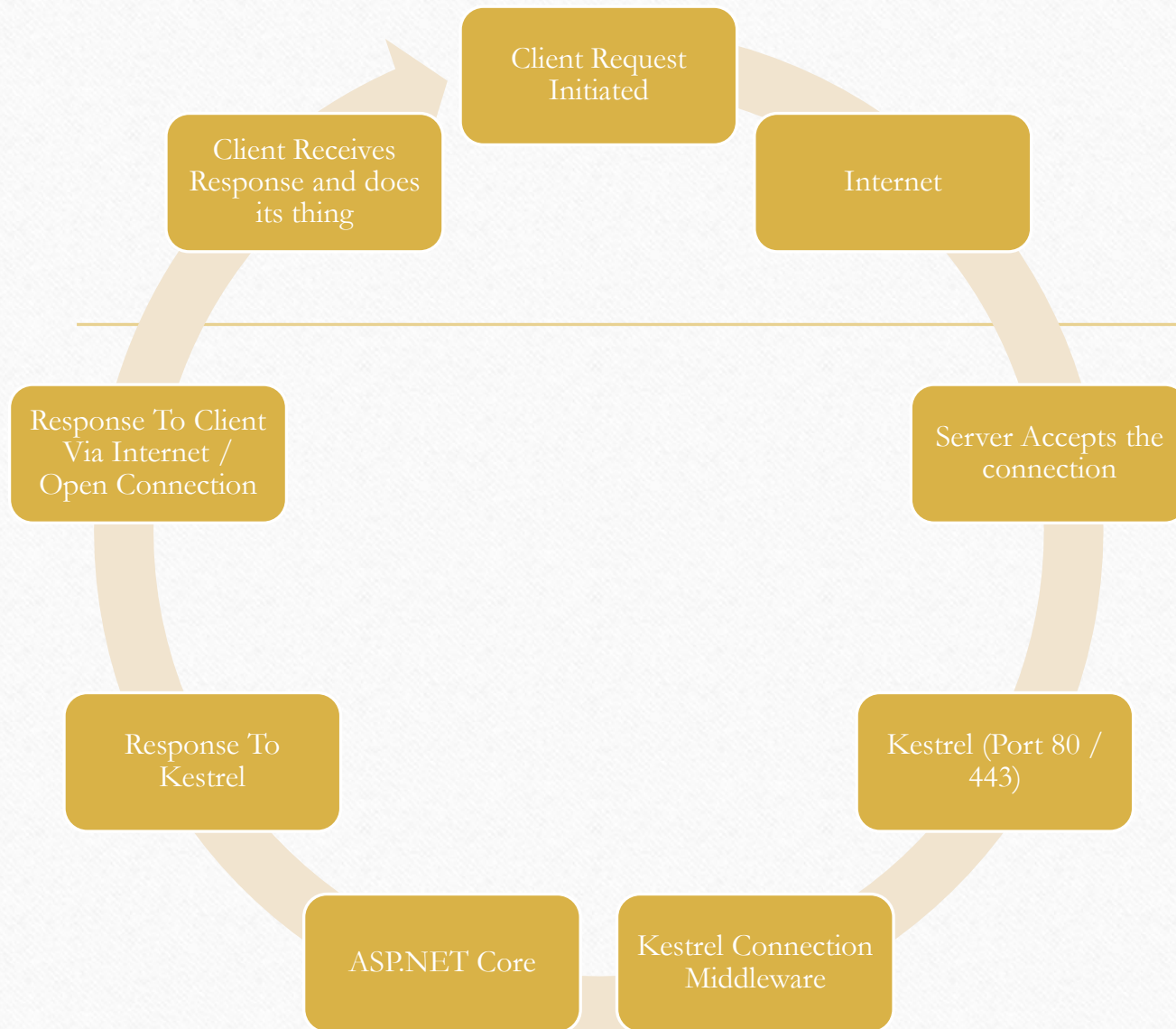
▼ Response Headers (141 B)

```
HTTP/1.1 200 OK
Date: Tue, 08 Jun 2021 11:06:26 GMT
Content-Type: text/html; charset=utf-8
Server: Kestrel
Transfer-Encoding: chunked
```

HTML

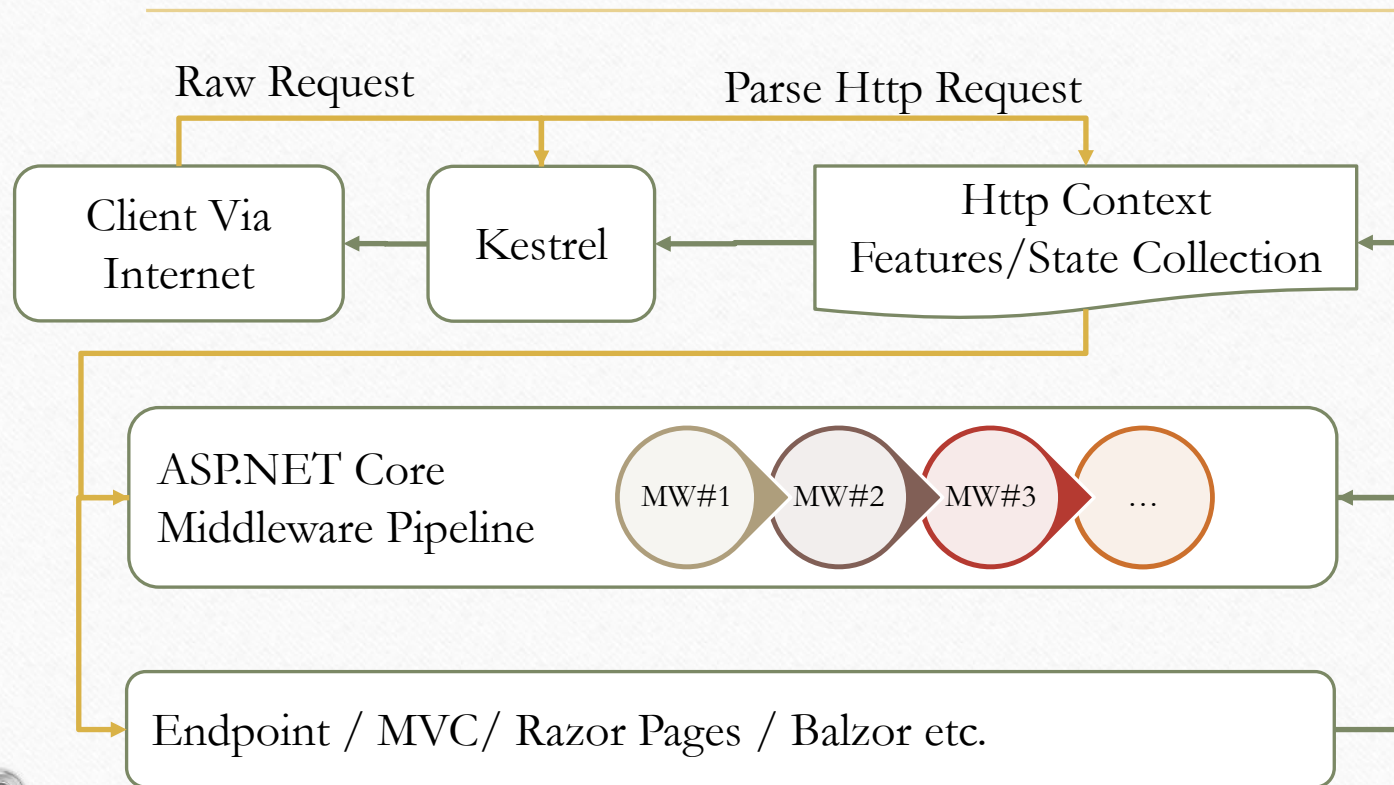
```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Photo Plash</title>
8   <link rel="stylesheet" href="/node_modules/bulma/css/bulma.min.css">
9   <link rel="stylesheet" href="/node_modules/@fortawesome/fontawesome-free/css/all.min.css">
10
11   <link rel="stylesheet" href="/css/site.css">
12 </head>
13
14 <body>
15   <nav class="navbar title is-6 has-text-white-bis has-background-black" role="navigation"
16     aria-label="main navigation">
17     <div class="navbar-brand">
18       <a class="navbar-item" href="/">
19         <p>
20           <i class="fa fa-paint-brush" aria-hidden="true"></i> Photo Plash
21         </p>
22     </div>
```

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	localhost:5000	/	BrowserTabChil...	html	5.22 KB	5.09 KB
304	GET	localhost:5000	01.jpg	img	jpeg	cached	30.68 KB
304	GET	localhost:5000	02.jpg	img	jpeg	cached	34.46 KB
304	GET	localhost:5000	67.jpg	img	jpeg	cached	36.07 KB
304	GET	localhost:5000	71.jpg	img	jpeg	cached	13.57 KB
200	GET	localhost:5000	favicon.ico	FaviconLoader.j...	x-icon	cached	16.56 KB



How Does Kestrel
Process a http/s
Request?

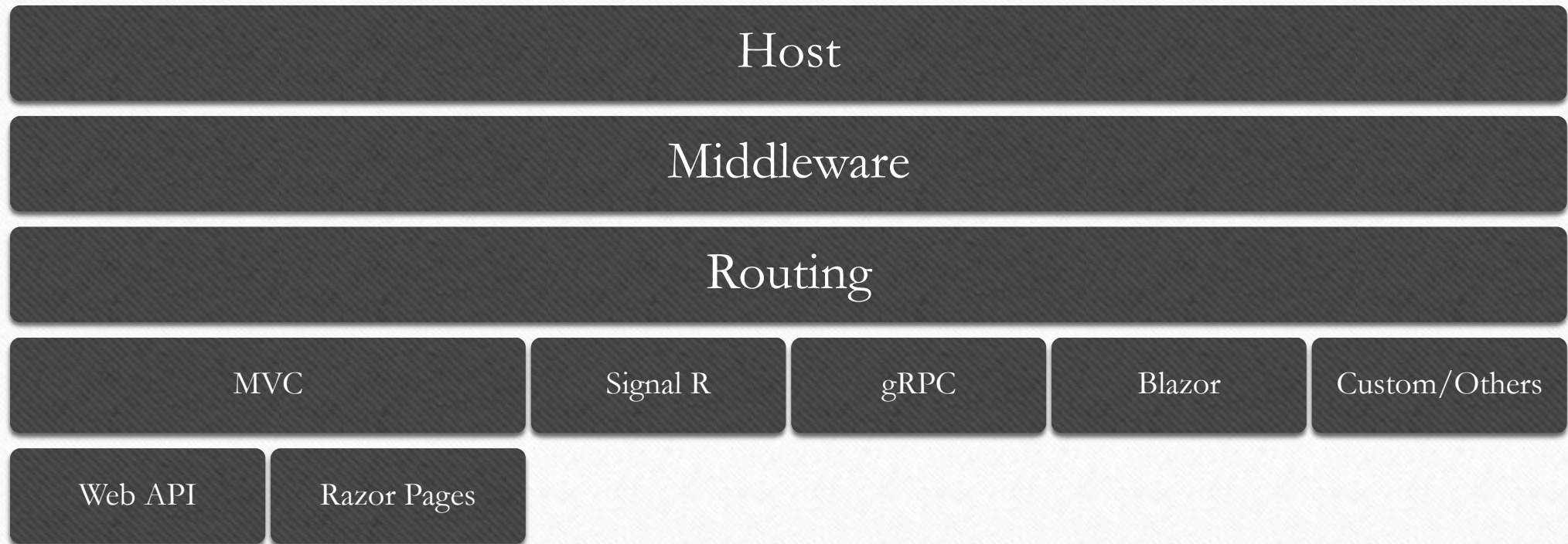
ASP.NET Core Request Response Pipeline



Let's Dig Into Kestrel
& ASP.NET Core
Request Processing

Request
Response

ASP.NET Core Architecture



The background of the slide is a photograph of an industrial facility, likely a refinery or chemical plant. It features a complex network of white and grey pipes, some with insulation. A prominent red handwheel valve is visible on the right side. The scene is brightly lit, suggesting daylight.

End of Session 1

Demo #1

- Start Building WebApp From Console App
- Host a static Html file
- Style using Raw CSS
- Style using Bulma



LET'S
DO IT!

Hands On Do It Yourself

- Start Building WebApp From Console App
- Host a static Html file
- Style using Raw CSS
- Style using Bulma

Demo #2

- Add MVC
- Add Razor Pages
- Error Handling / Environment
- Simple Navigation Menu



LET'S
DO IT!

Hands On Do It Yourself

- Add MVC
- Add Razor Pages
- Error Handling / Environment
- Simple Navigation Menu

Demo #3

- Basic Forms – Send Mail
- Tag Helpers
- Form Validation on Client using JQuery
- Null Mail Service
- Dummy Data Repositories (Use Dummy Sqlite for Development and Sql Server for Others)



LET'S
DO IT!

Hands On Do It Yourself

- Basic Forms – Send Mail
- Tag Helpers
- Form Validation Client side using jquery
- Null Mail Service

Best Practices

- Builder Pattern
- Dependency Inversion / IOC Containers / Dependency Injection
- Repository Pattern
- Singleton / Transient / Scoped Objects
- DI's role in Unit Testing
- How to approach S in SOLID





Thank You

Manoj Bhargavan