ASP.NET Core 6 Web API Fundamentals

Getting Acquainted with ASP.NET Core

Kevin Dockx
Architect

@KevinDockx https://www.kevindockx.com

ASP.NET Core 6 Web API Fundamentals

Version Check

Version Check

This version was created by using:

- ASP.NET Core 6.0
- Entity Framework Core 6.0
- .NET 6.0
- Visual Studio 2022

Version Check

This course is 100% applicable to:

- ASP.NET Core 6.x, 7.x
- Entity Framework Core 6.x, 7.x
- .NET 6.x, 7.x

Relevant Notes

New course versions are regularly released:

- https://app.pluralsight.com/profile/author/kevin-dockx

Coming Up

Course structure

Prerequisites, frameworks and tooling

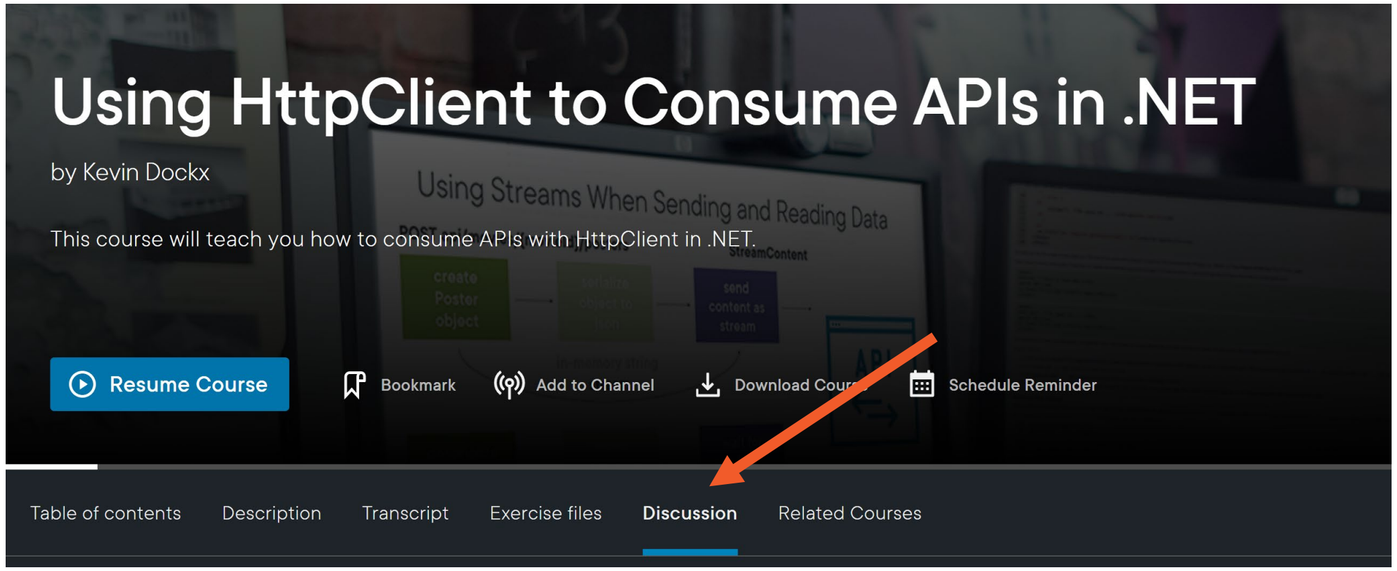
ASP.NET Core 6: the big picture

Starting an ASP.NET Core 6 project

Middleware and the request pipeline

Discussion tab on the course page

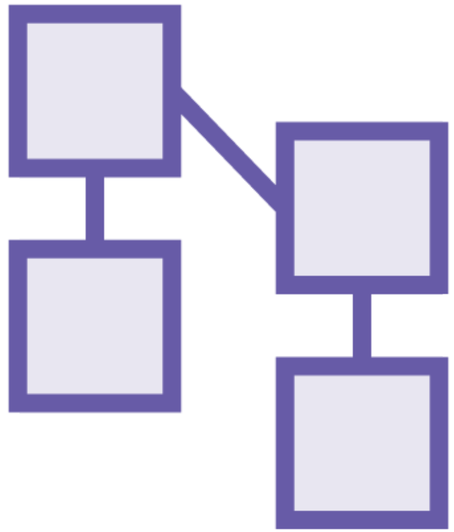
Twitter: @KevinDockx



(course shown is one of my other courses, not this one)

About the Course Structure

From introducing ASP.NET Core 6...



... over building the API ...

... and using Entity Framework Core 6...

... to common resource concerns, security, versioning and documentation

Course Prerequisites

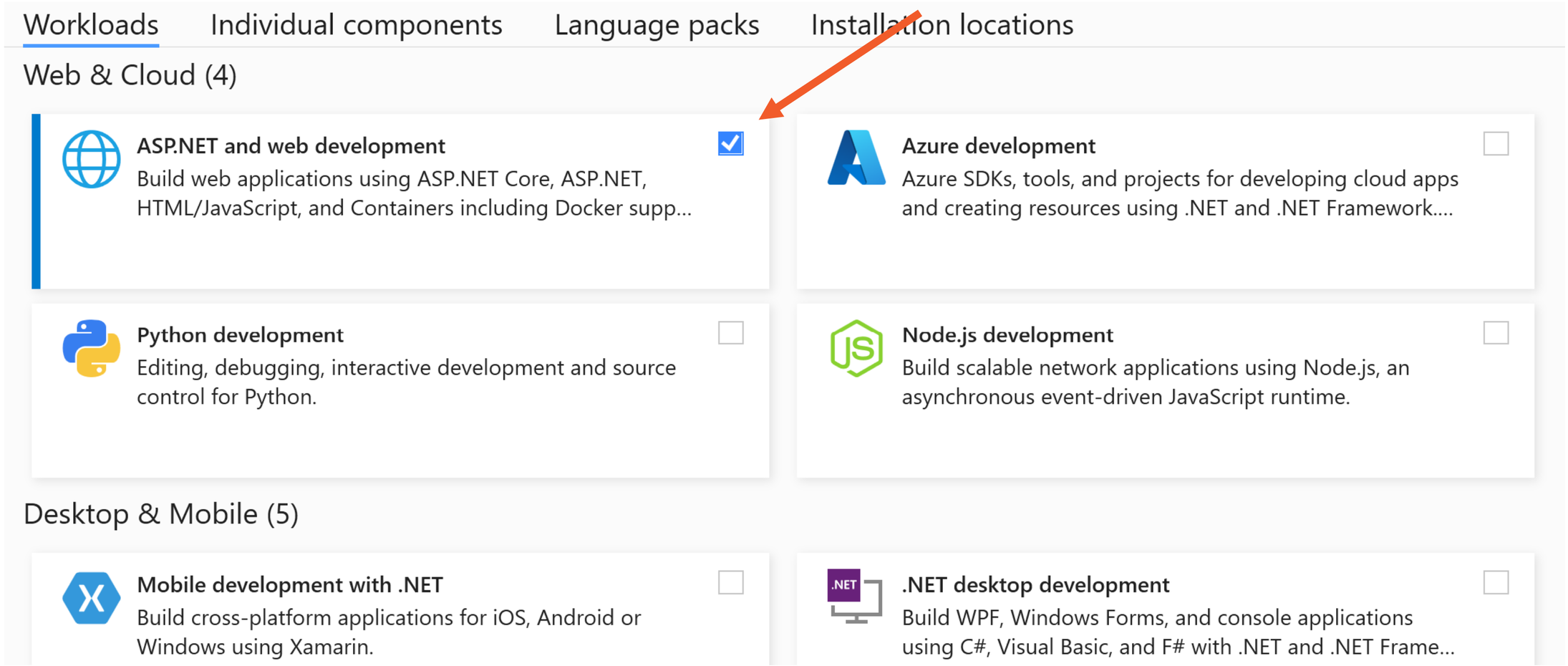
Beginner course, no specific knowledge required in regards to

- .NET 6
- ASP.NET Core 6
- Entity Framework 6

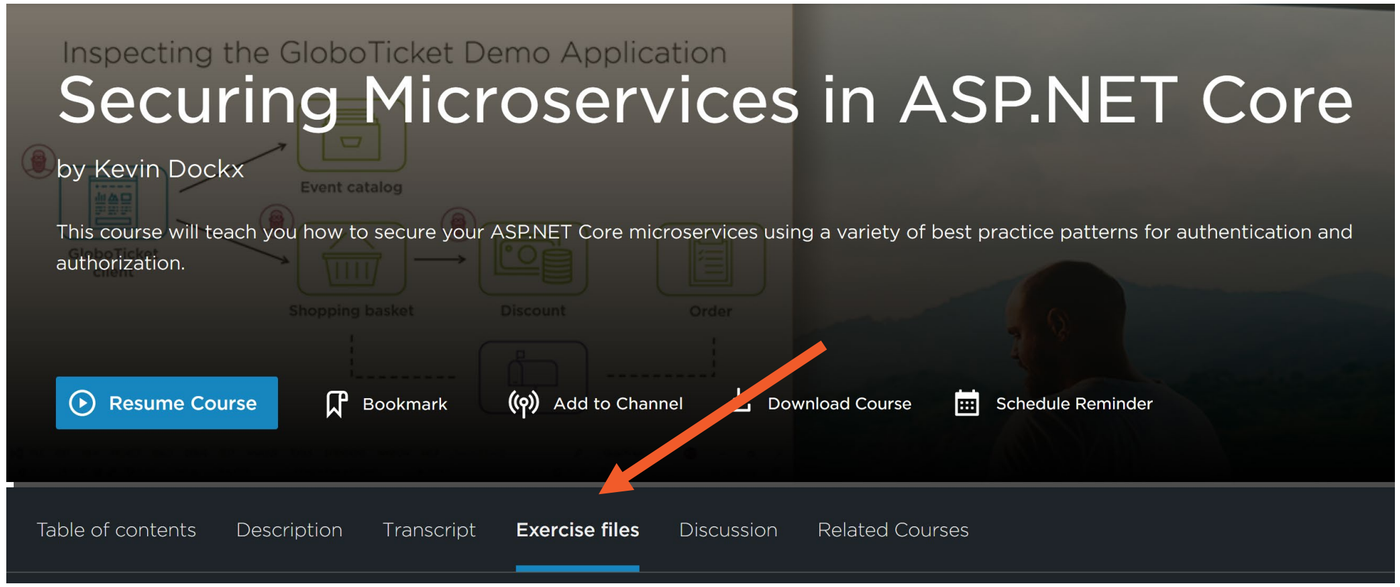
Knowledge of C# is required

- C# 10 is used in this course

Installing Visual Studio



Exercise files tab on the course page



(course shown is one of my other courses, not this one)

ASP.NET Core

A cross-platform, high-performance, open-source framework for building modern, cloud-enabled, Internet-connected apps

ASP.NET Core: The Big Picture

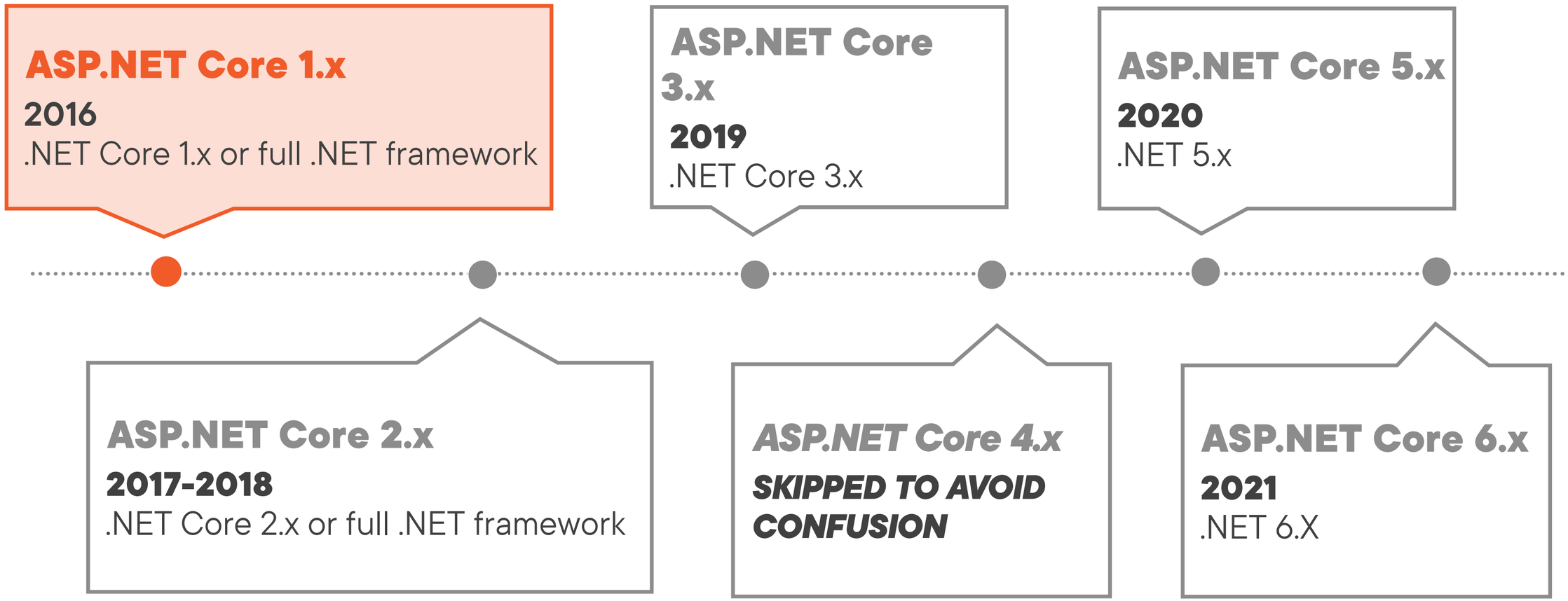
Build web apps and services, IoT apps and mobile backends

Develop on Windows, Linux and macOS

Deploy to the cloud or on-premises

Run on .NET (or .NET Core, depending on the version)

ASP.NET Core History



ASP.NET Core History

ASP.NET Core 1.x
2016
.NET Core 1.x or full .NET framework

ASP.NET Core 2.x
2017-2018
.NET Core 2.x or full .NET framework

ASP.NET Core 3.x
2019
.NET Core 3.x

ASP.NET Core 4.x
SKIPPED TO AVOID CONFUSION

ASP.NET Core 5.x
2020
.NET 5.x

ASP.NET Core 6.x
2021
.NET 6.X

ASP.NET Core History

ASP.NET Core 1.x
2016
.NET Core 1.x or full .NET framework

ASP.NET Core 2.x
2017-2018
.NET Core 2.x or full .NET framework

ASP.NET Core 3.x
2019
.NET Core 3.x

ASP.NET Core 4.x
SKIPPED TO AVOID CONFUSION

ASP.NET Core 5.x
2020
.NET 5.x

ASP.NET Core 6.x
2021
.NET 6.X

ASP.NET Core History

Full .NET framework (4.8)
No new development

.NET Core vNext (“4.x”)
Useful functionality from the full .NET framework ported to increase performance and support cross-platform development

ASP.NET Core History

ASP.NET Core 1.x
2016
.NET Core 1.x or full .NET framework

ASP.NET Core 2.x
2017-2018
.NET Core 2.x or full .NET framework

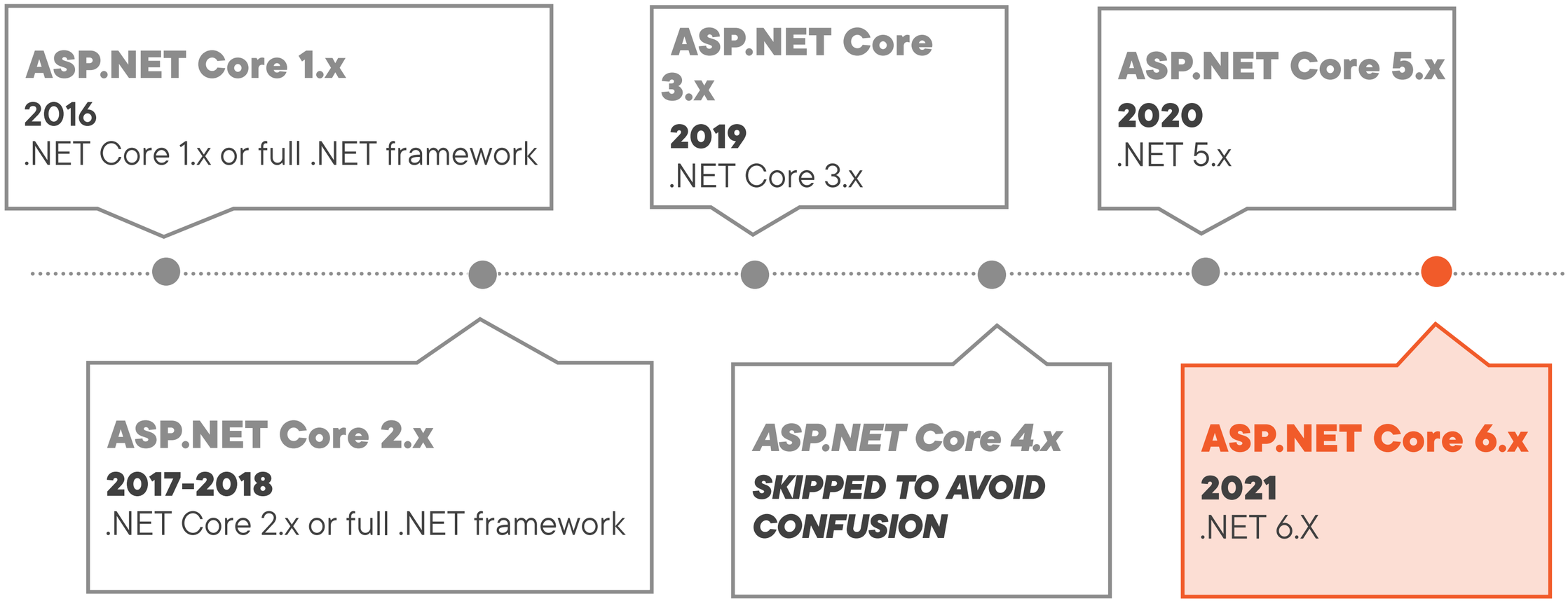
ASP.NET Core 3.x
2019
.NET Core 3.x

ASP.NET Core 4.x
SKIPPED TO AVOID CONFUSION

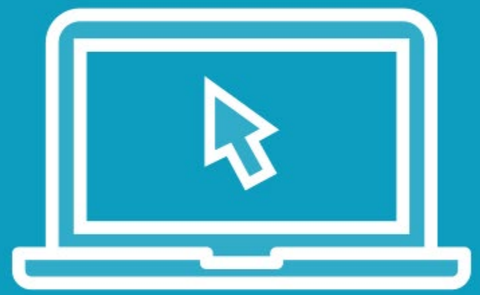
ASP.NET Core 5.x
2020
.NET 5.x

ASP.NET Core 6.x
2021
.NET 6.X

ASP.NET Core History

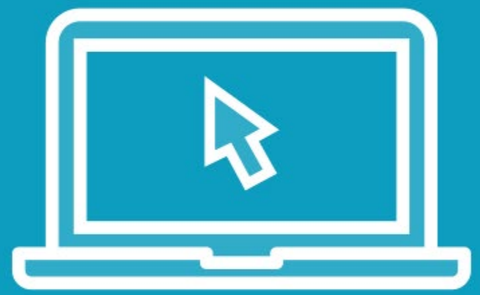


Demo



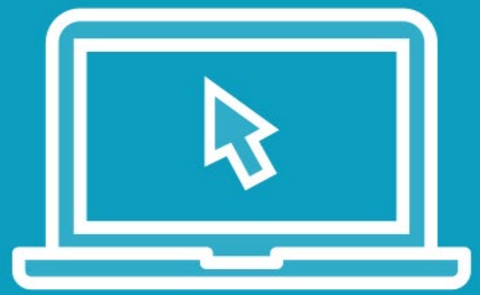
Creating and running a new ASP.NET Core project

Demo



Running an ASP.NET Core project using the CLI

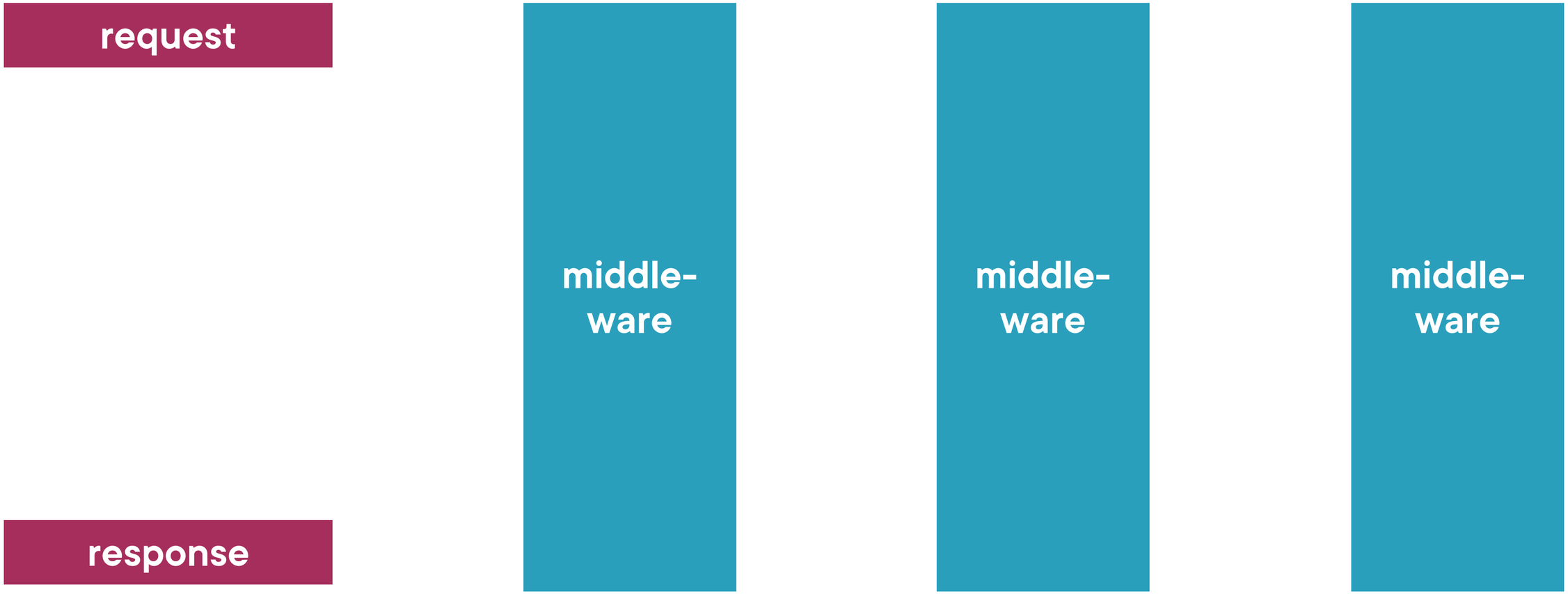
Demo



Inspecting the ASP.NET Core API project code structure

خطوط الأنابيب

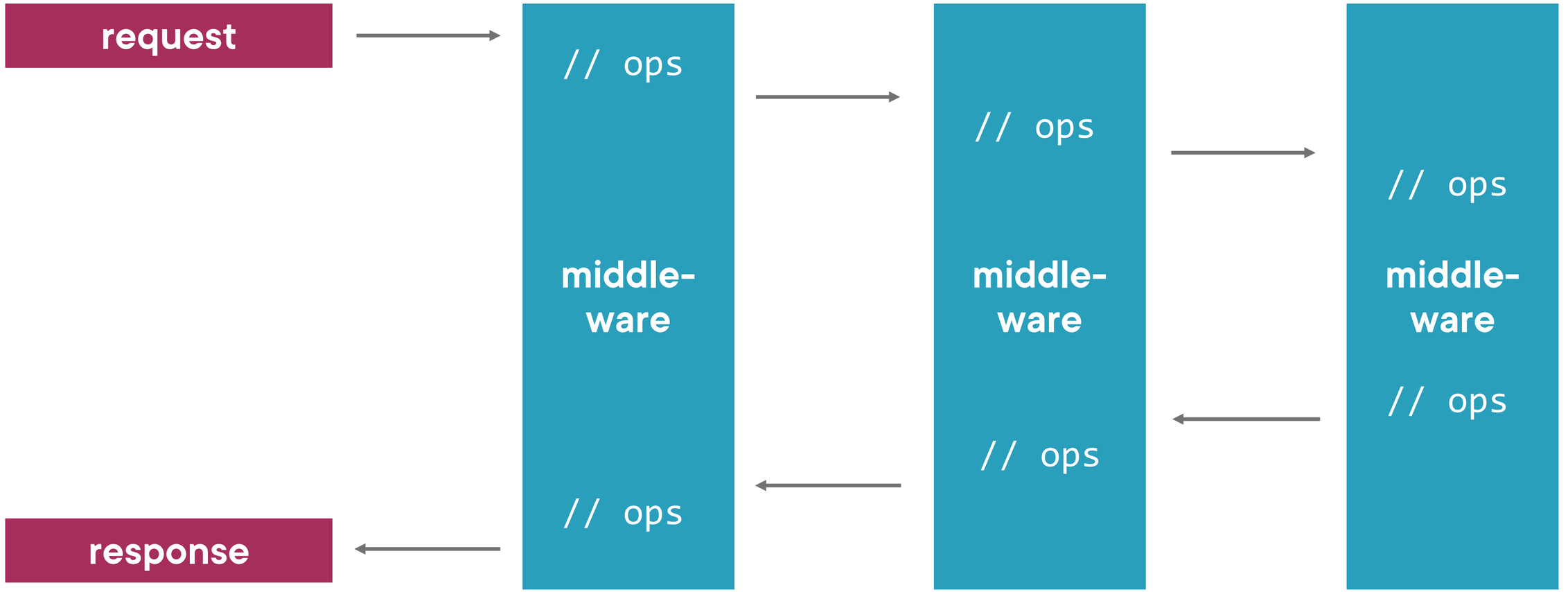
The ASP.NET Core Request Pipeline & Middleware



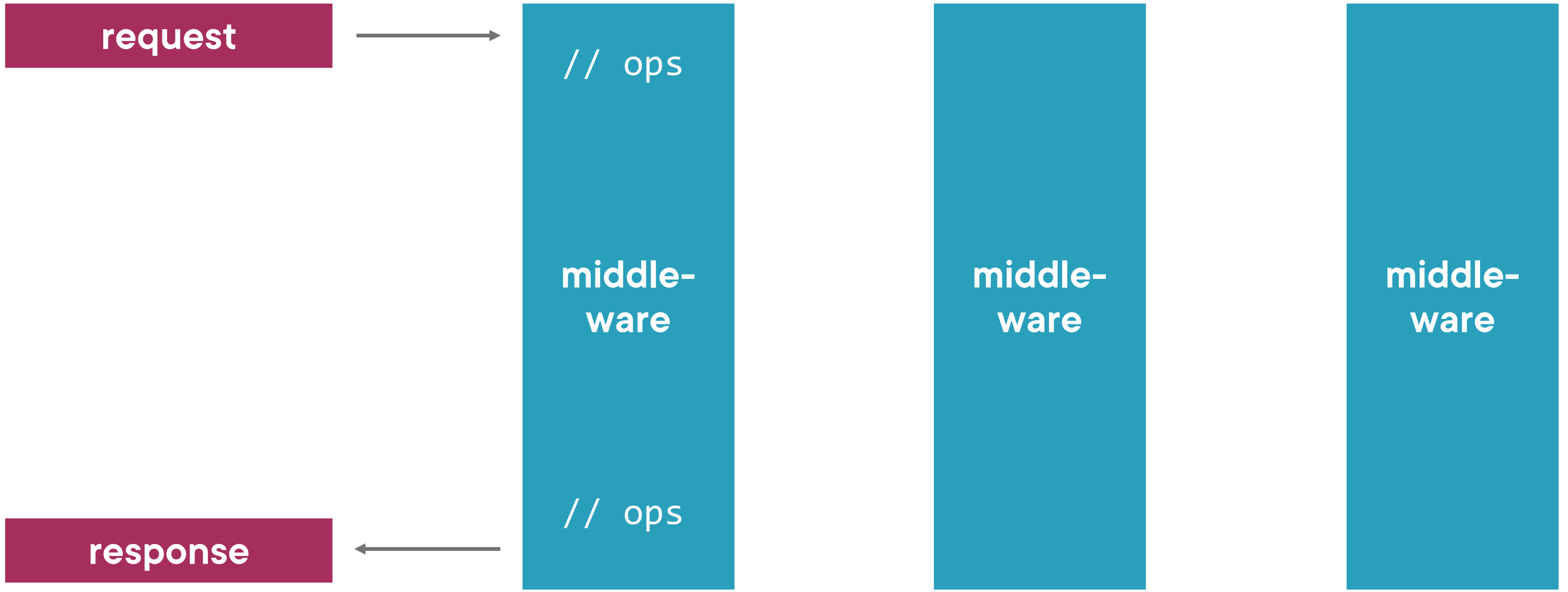
Middleware

Software components that are assembles into an application pipeline to handle requests and responses

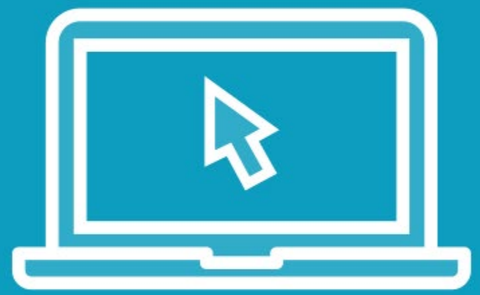
The ASP.NET Core Request Pipeline & Middleware



The ASP.NET Core Request Pipeline & Middleware

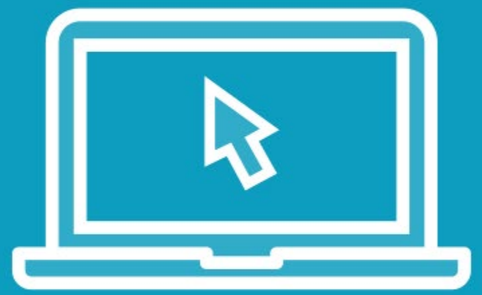


Demo



Configuring the ASP.NET Core request pipeline

Demo



Working with different environments

Summary

ASP.NET Core

- Cross-platform
- High-performance
- Open-source
- Modern, cloud-enabled, Internet-connected apps

Current version: ASP.NET Core 6, on .NET 6

Summary

`Program` class is the starting point of the application

- Generated `Main` method is responsible for configuring and running the application

Configure services by adding them to the services collection on the builder (`WebApplicationBuilder`)

Summary

After building the web application via the `WebApplicationBuilder`, configure the request pipeline by adding middleware to it

- Potentially, scope this middleware to different environments

Up Next:
Creating the API and Returning Resources