

# Introduction to ASP.NET Core

## CONTENTS

- 01 What is ASP.NET Core?
- 02 How ASP.NET Core Works?
- 03 Your First Application

## 1. What is ASP.NET Core?

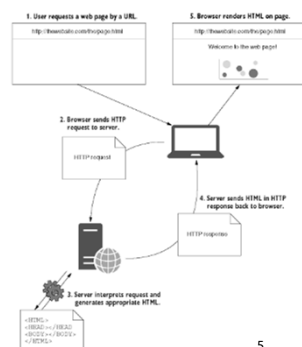
### What is ASP.NET Core?

- ASP.NET Core is a **cross-platform, open-source** application framework that you can use to build dynamic **web applications**.
- Selling points of ASP.NET Core:
  - modern, high-performance, open-source
  - familiar design patterns and paradigms
  - C#
  - cross-platform

4

### Primer on Web Application

- Web applications are apps running at web servers that you can view in a web browser.
- The figure shows how an HTTP request works.



5

### Dynamic Web Page

- it's perfectly possible to build a **static web application** without the use of a web framework, but its capabilities will be limited.
- The key to any modern web application is the ability to generate **dynamic web pages**.
- Dynamic web page may display different data depending on the current user, or it could display content submitted by users.
- Without a dynamic **framework**, it wouldn't be possible to log in to websites or to display any sort of personalized data on a page.

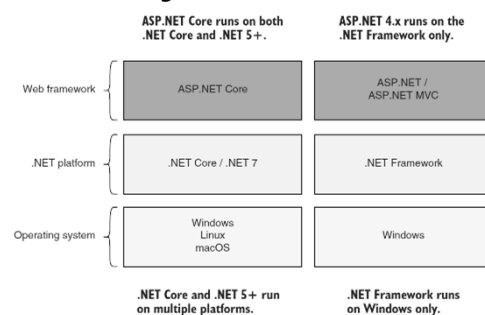
6

## An Example of Dynamic Web Page



7

## Relations among ASP.NET Core, ASP.NET, .NET



8

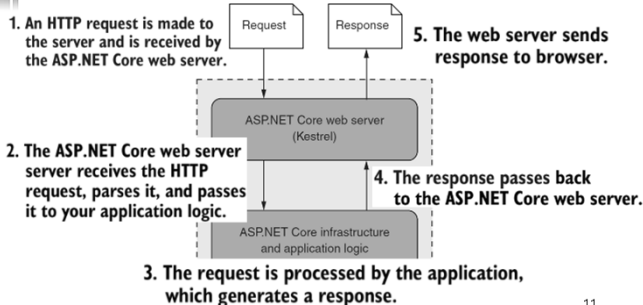
## 2. How ASP.NET Core Work?

### What are ASP.NET Core Applications?

- ASP.NET Core applications are **console applications** that self-host a web server and handle requests directly.
- Kestrel is a fast, cross-platform HTTP server on which ASP.NET Core can run.
  - It receives raw requests;
  - It constructs an internal representation of the data, an `HttpContext` object, which the rest of the application can use.

10

### How Does ASP.NET Core Process a Request?



11

### HttpContext

- The **HttpContext** constructed by the web server is used by the application as a sort of storage box for a single request.
- Your application can use the details stored in `HttpContext` to generate an appropriate response to the request:
  - generate some HTML;
  - return an "access denied" message;
  - send an email

12

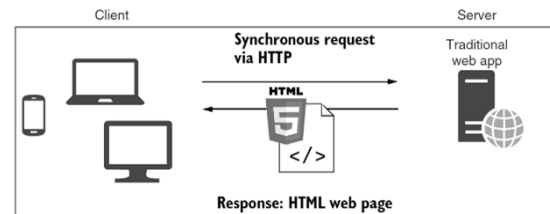
## Paradigms of ASP.NET Core

- ASP.NET Core provides a generalized web framework that can be used to build a wide variety of applications:
  - Traditional page-based, server-side-rendered Web applications
  - REST or HTTP API server
  - RPC-style service
  - Browser-based single-page application

13

## Traditional Page-based, Server-side-rendered Web Applications

- Razor pages
- MVC controllers



14

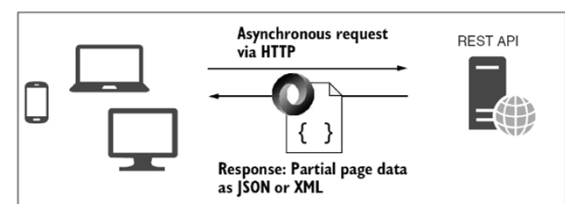
## Example of a Server-side-rendered Web Application



15

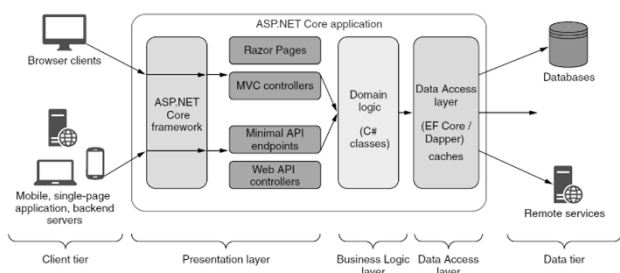
## REST or HTTP API Server

- Minimal API
- Web API



16

## A Typical ASP.NET Core Application



17