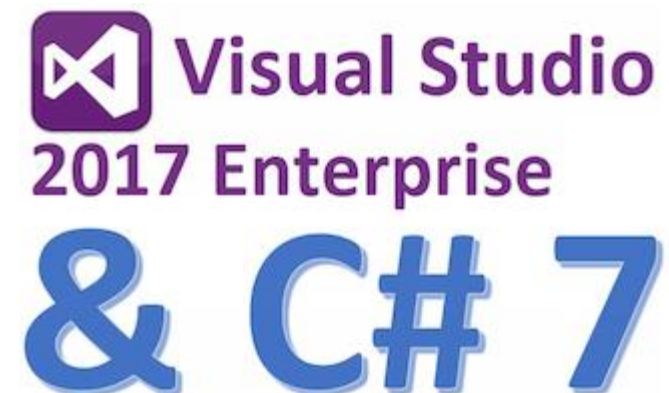


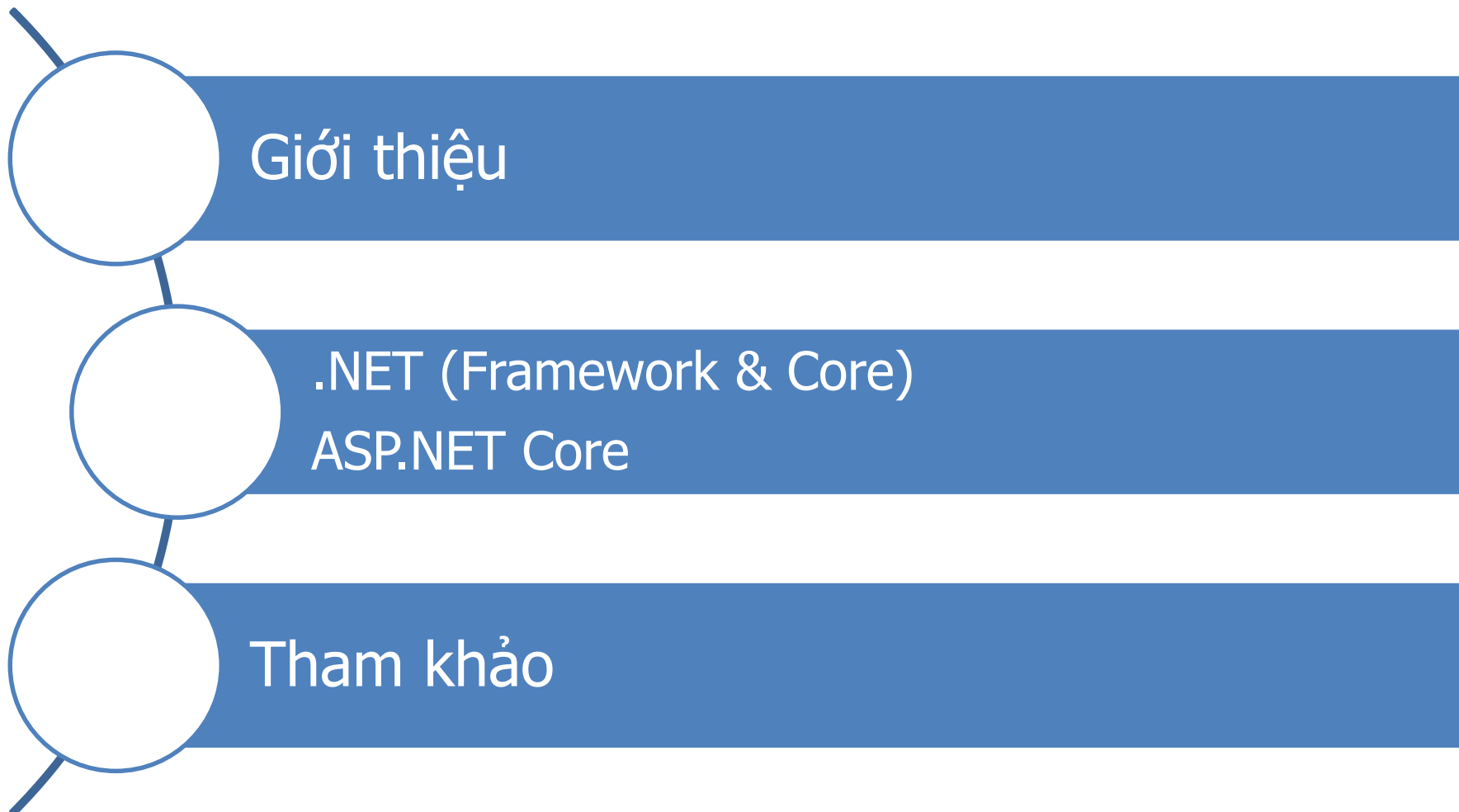
ASP.NET Core 2.x

The Future of Web Apps



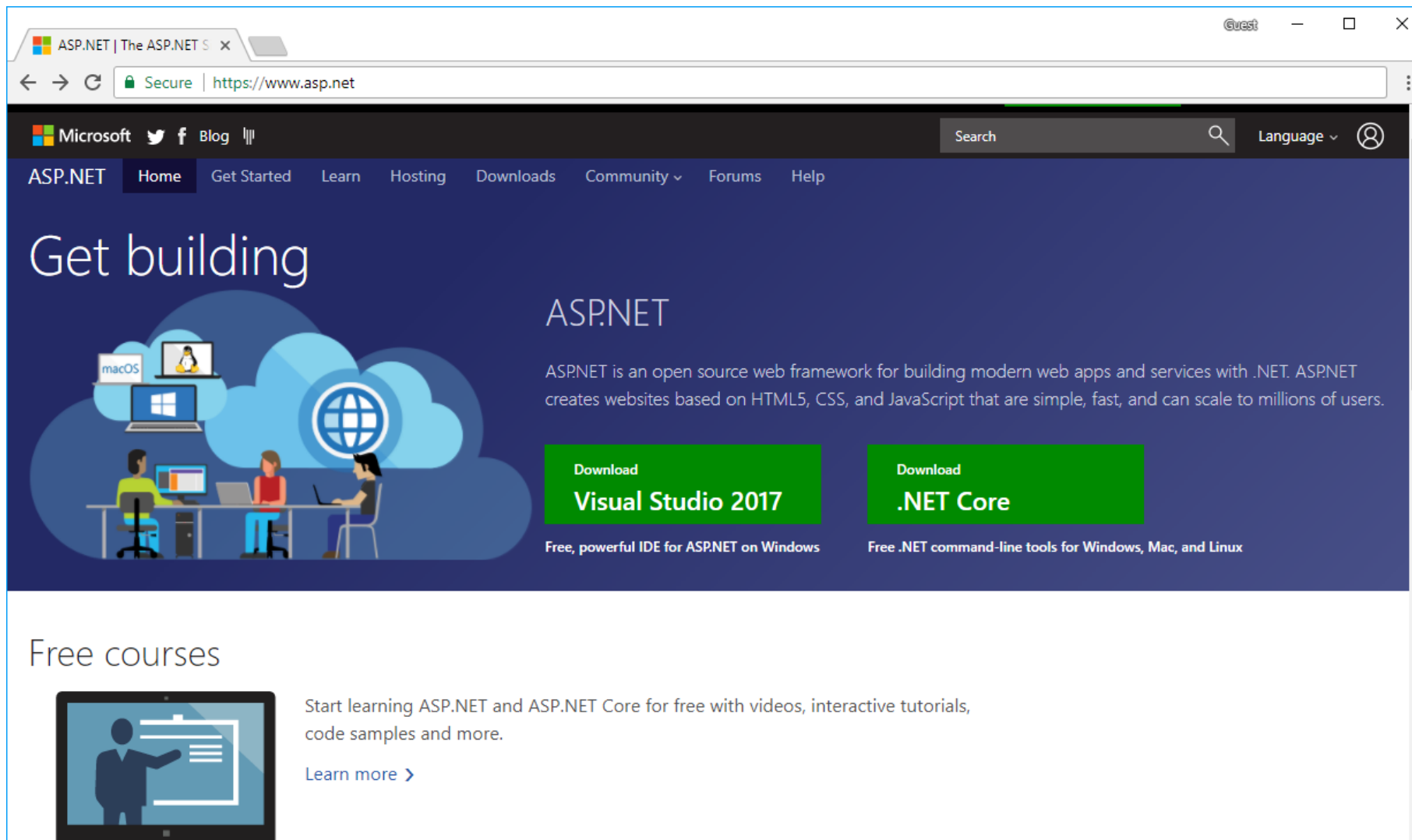
Lương Trần Hy Hiền
hyhien@gmail.com – 0989.366.990





GIỚI THIỆU

Info and Downloads: <http://www.asp.net/>



The screenshot shows the ASP.NET website homepage. The browser window has a single tab titled "ASP.NET | The ASP.NET S...". The address bar shows "Secure | https://www.asp.net". The page features a dark blue header with the Microsoft logo, social media links, a search bar, and a language selector. The main navigation bar includes links for ASP.NET, Home, Get Started, Learn, Hosting, Downloads, Community, Forums, and Help. The main content area has a large "Get building" section with an illustration of people working on laptops. Below this, there are two green buttons for downloading Visual Studio 2017 and .NET Core. The bottom section is titled "Free courses" and includes an illustration of a person presenting at a screen, with text encouraging users to start learning ASP.NET and ASP.NET Core for free.

ASP.NET | The ASP.NET S x

Secure | https://www.asp.net

Microsoft Blog

ASP.NET Home Get Started Learn Hosting Downloads Community Forums Help

Get building

ASP.NET

ASP.NET is an open source web framework for building modern web apps and services with .NET. ASP.NET creates websites based on HTML5, CSS, and JavaScript that are simple, fast, and can scale to millions of users.

Download Visual Studio 2017

Free, powerful IDE for ASP.NET on Windows

Download .NET Core

Free .NET command-line tools for Windows, Mac, and Linux

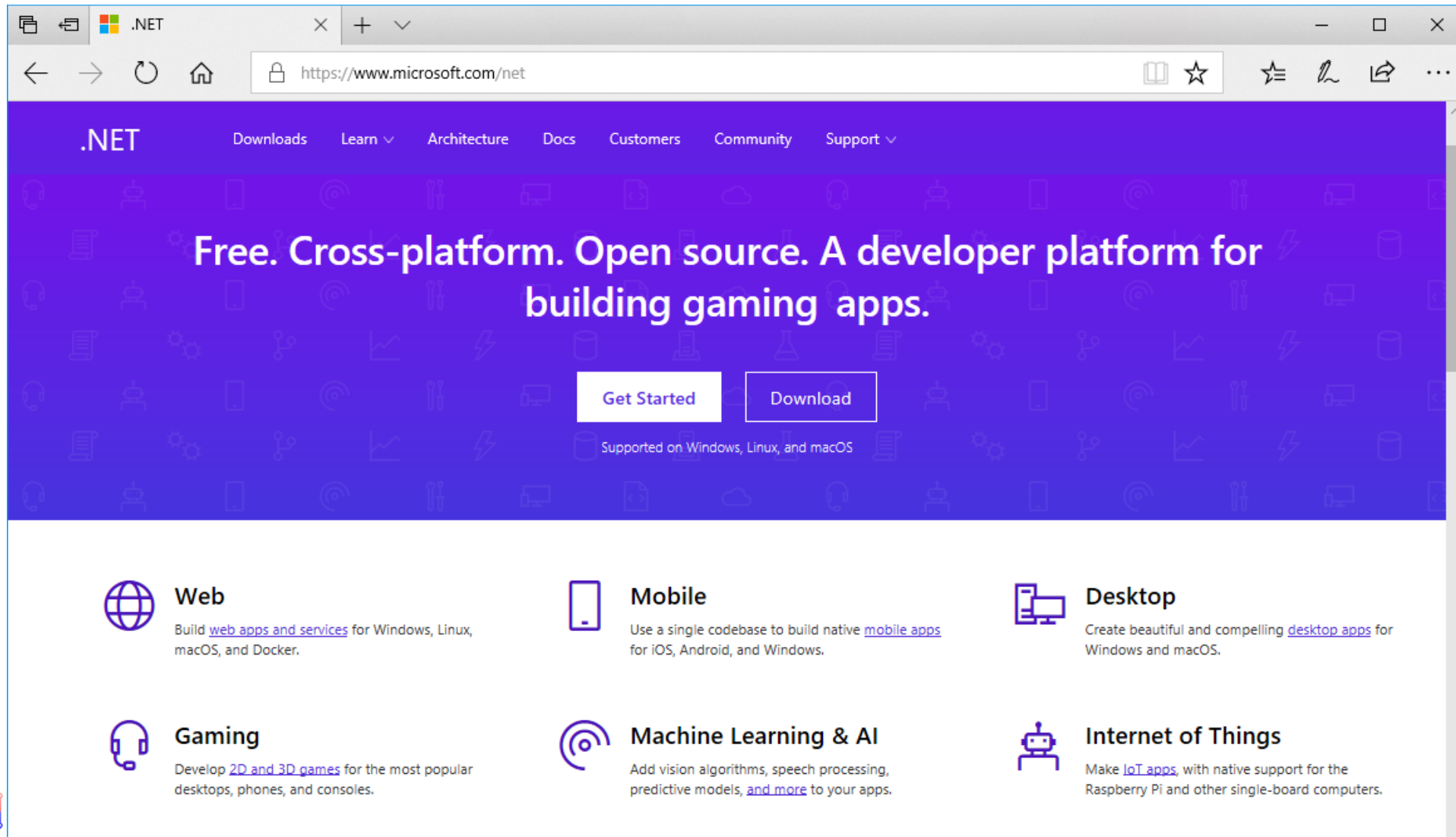
Free courses

Start learning ASP.NET and ASP.NET Core for free with videos, interactive tutorials, code samples and more.

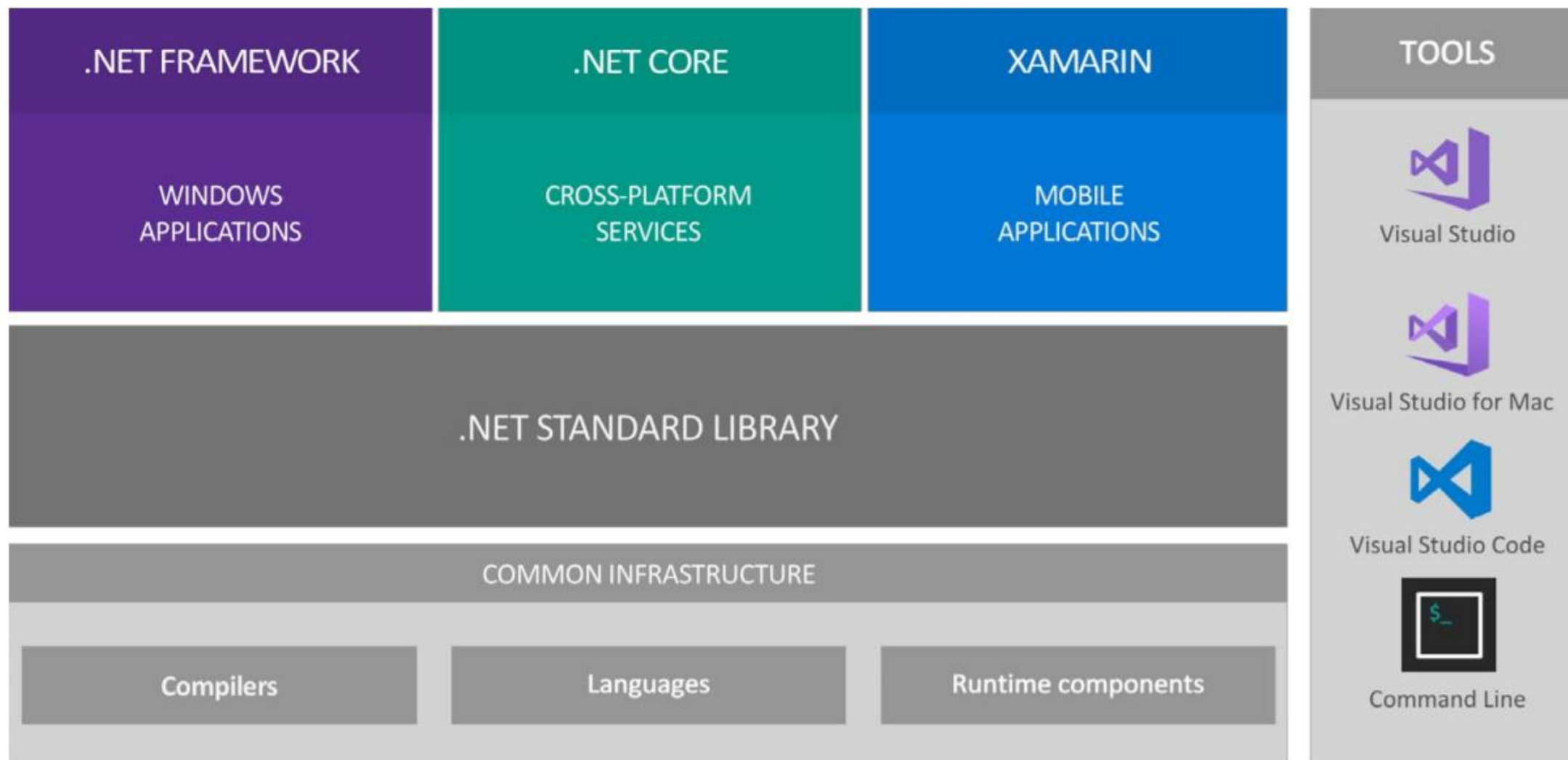
Learn more >

.NET Core for Cross-Platform Dev

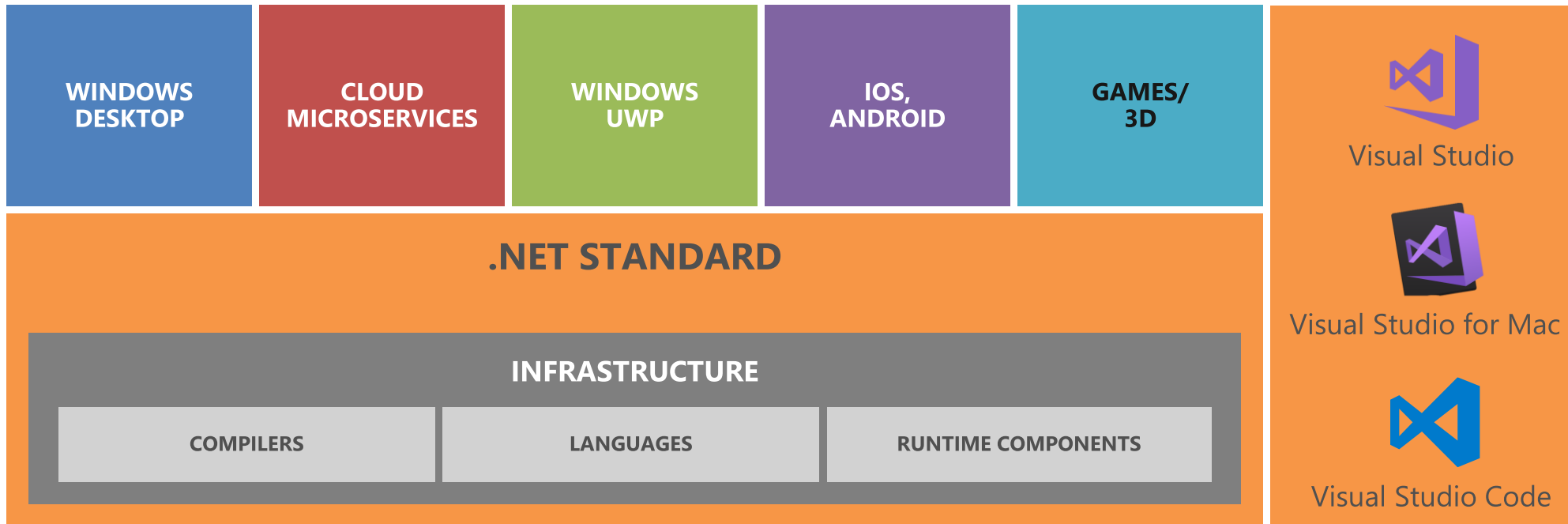
.NET Core: <https://www.microsoft.com/net>



.NET Standard in 2018



.NET Standard

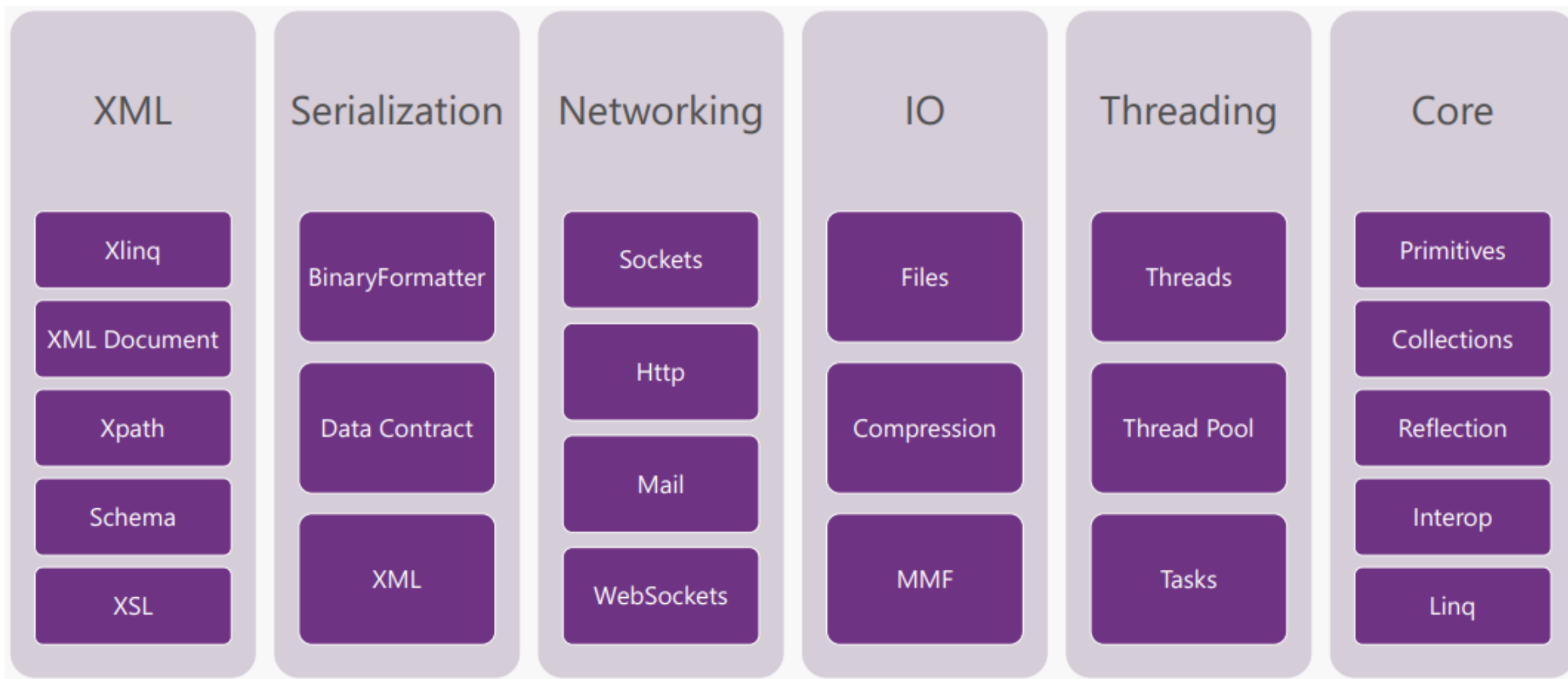


.NET Standard allows sharing code, binaries, and skills between .NET client, server, and all flavors

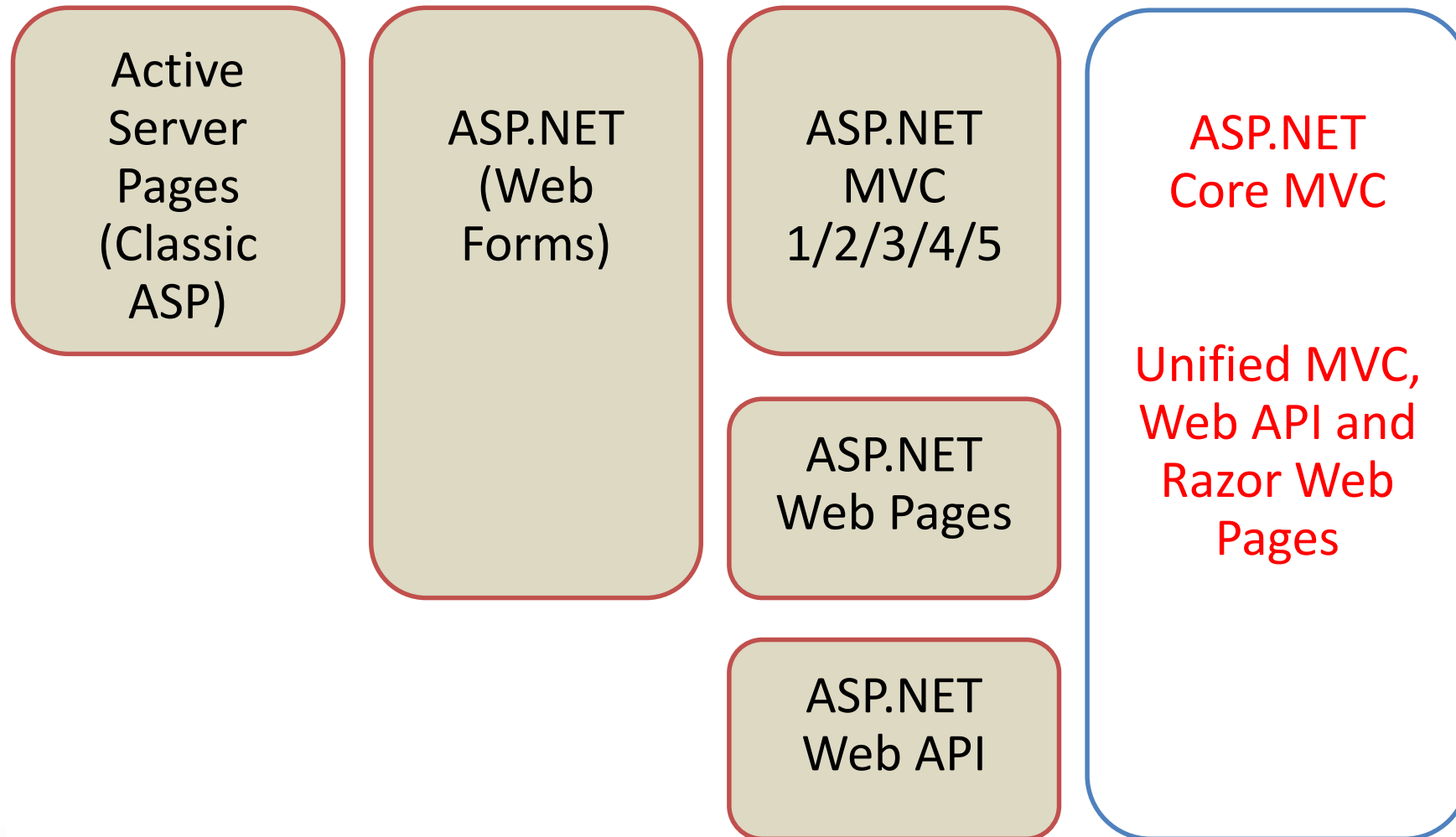
.NET Standard provides a specification for any platform to implement

All .NET runtimes provided by Microsoft implement the standard

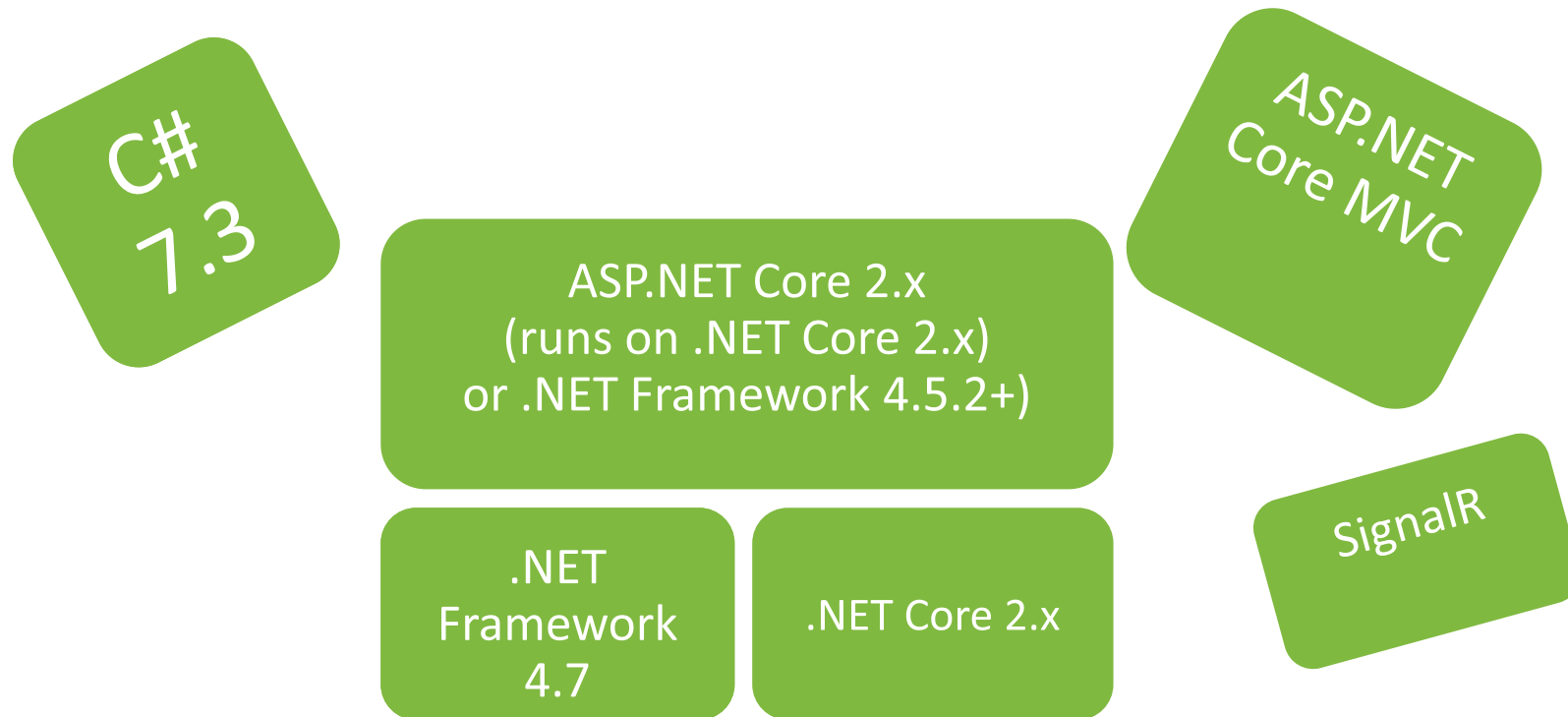
APIs in .NET Standard 2.0



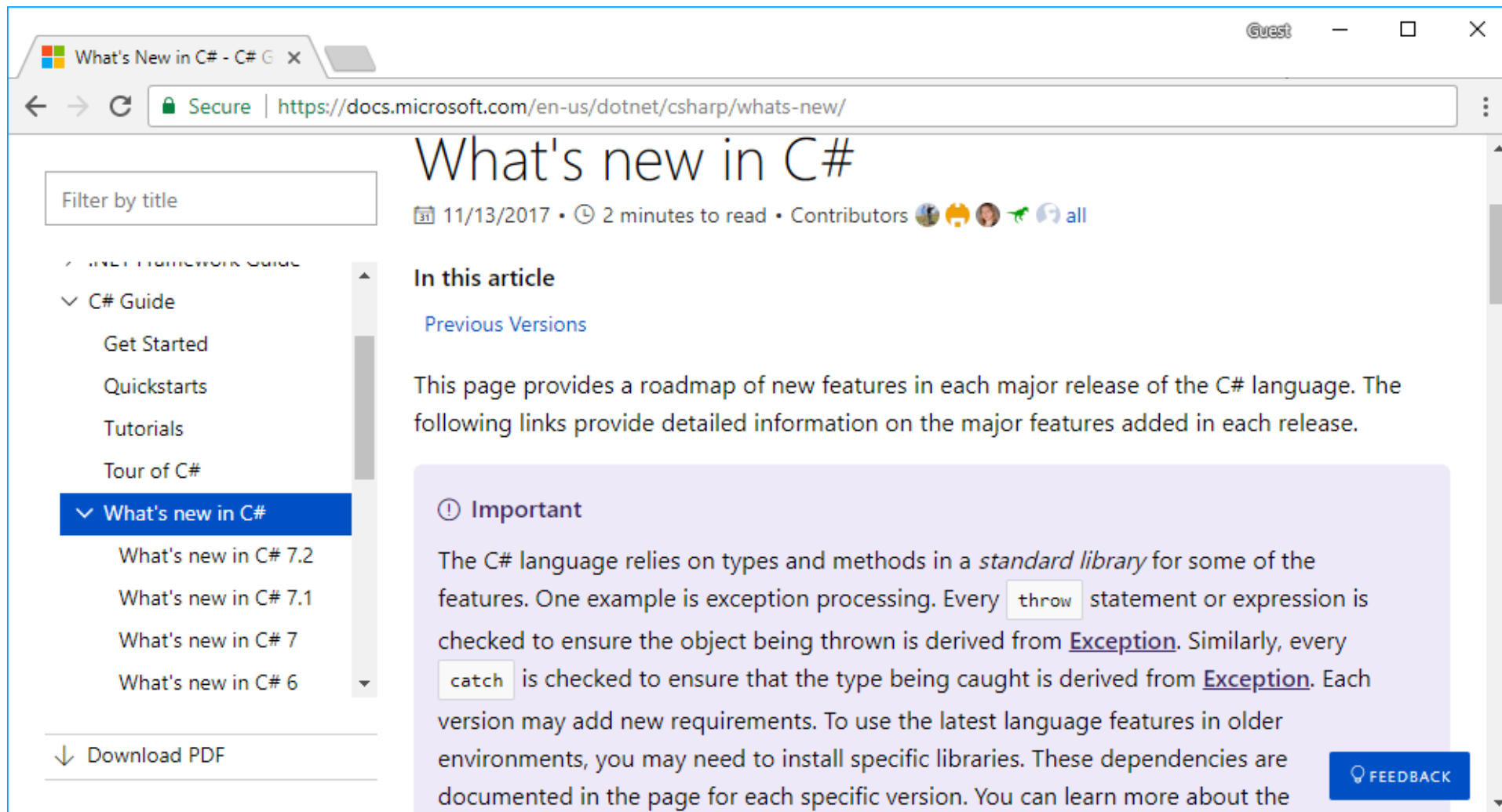
Sự phát triển của ASP và ASP .NET



Names & Version Numbers



C# 7.x in VS2017



The screenshot shows a web browser window displaying the 'What's new in C#' article. The browser's address bar shows the URL <https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/>. The article title is 'What's new in C#', dated 11/13/2017, with a 2-minute read time. The left sidebar contains a navigation menu with 'What's new in C#' selected. The main content area includes an 'Important' section discussing exception processing and a 'FEEDBACK' button.

What's new in C#

11/13/2017 • 2 minutes to read • Contributors

Filter by title

- NET Framework Guide
- ✓ C# Guide
 - Get Started
 - Quickstarts
 - Tutorials
 - Tour of C#
 - ✓ **What's new in C#**
 - What's new in C# 7.2
 - What's new in C# 7.1
 - What's new in C# 7
 - What's new in C# 6

Download PDF

In this article

[Previous Versions](#)

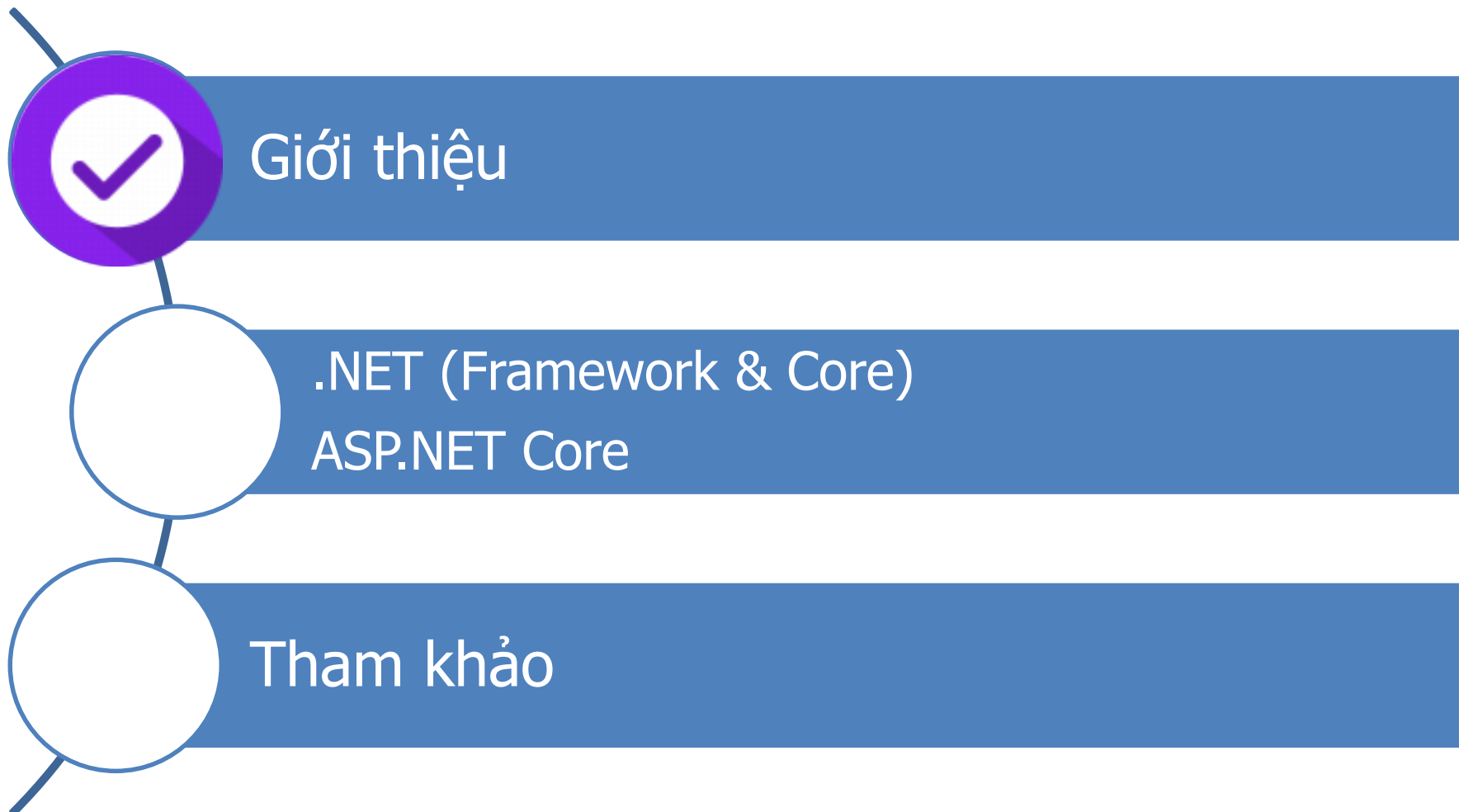
This page provides a roadmap of new features in each major release of the C# language. The following links provide detailed information on the major features added in each release.

Important

The C# language relies on types and methods in a *standard library* for some of the features. One example is exception processing. Every `throw` statement or expression is checked to ensure the object being thrown is derived from Exception. Similarly, every `catch` is checked to ensure that the type being caught is derived from Exception. Each version may add new requirements. To use the latest language features in older environments, you may need to install specific libraries. These dependencies are documented in the page for each specific version. You can learn more about the

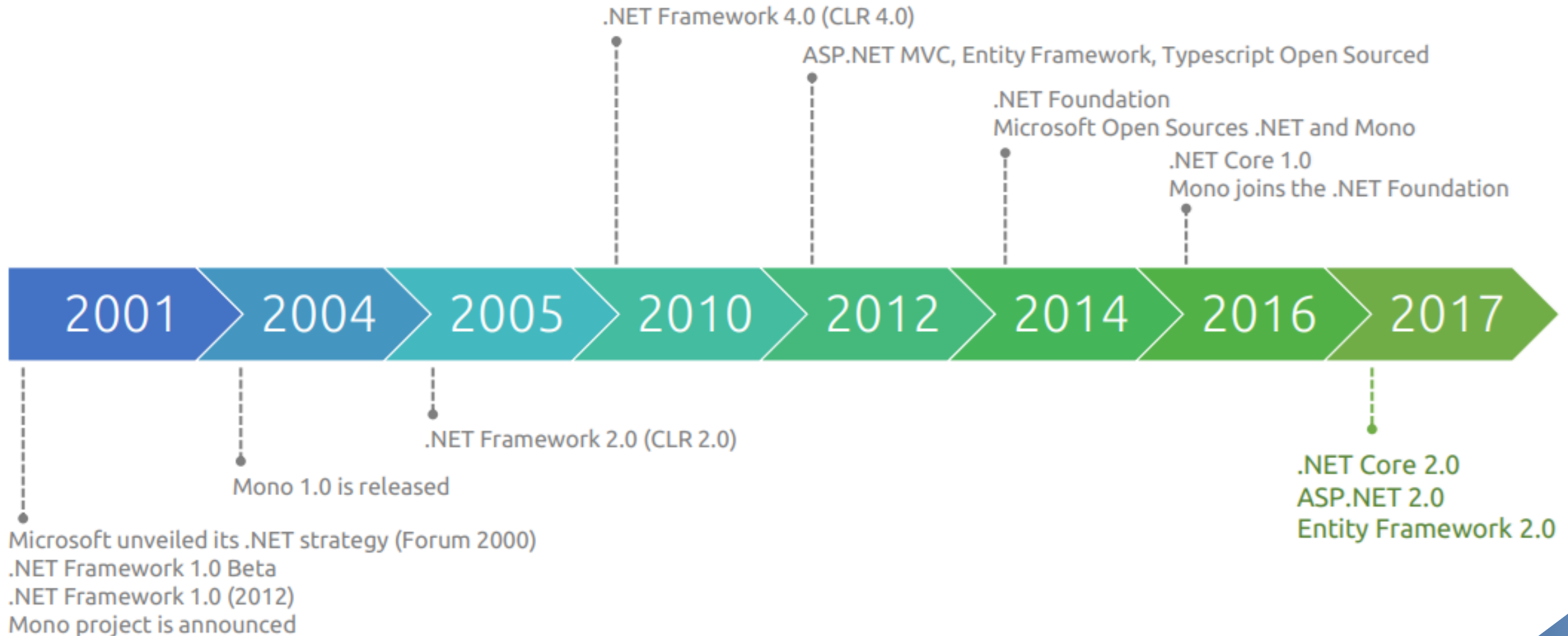
FEEDBACK

<https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/>



.NET FRAMEWORK VS .NET CORE

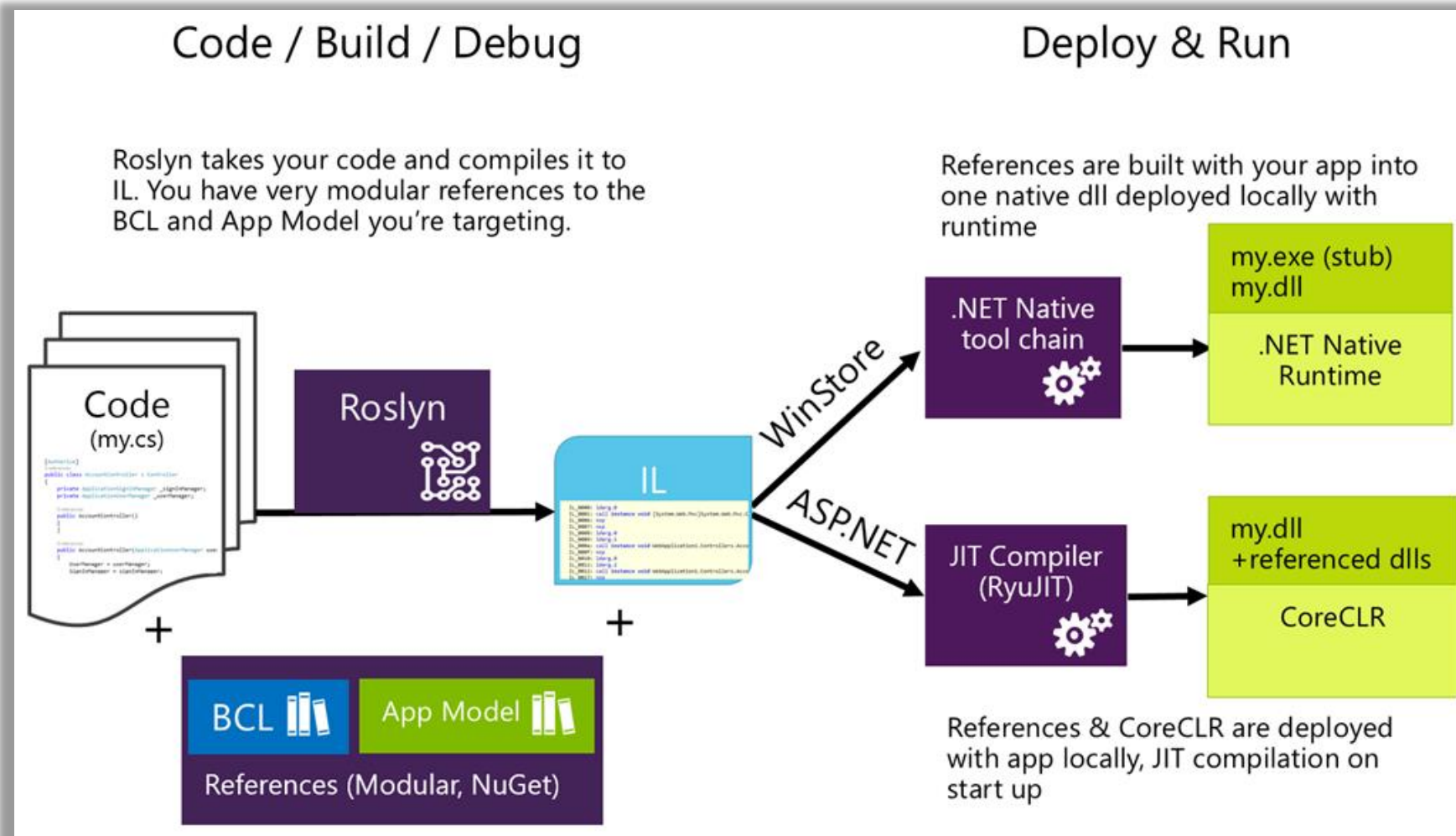
History of .NET Framework



Tại sao là .NET Core?

- ☐ **Open source**
- ☐ **Cross-platform**
- ☐ **Compatible**
- ☐ **Flexible deployment**
- ☐ **Command-line tools**

Compilation Process



Chuẩn bị môi trường

☐ Cài đặt SDK

<https://www.microsoft.com/net/download/core>

☐ Cài Visual Studio 2017 (Optional)

☐ Kiểm tra version:

`dotnet --version`

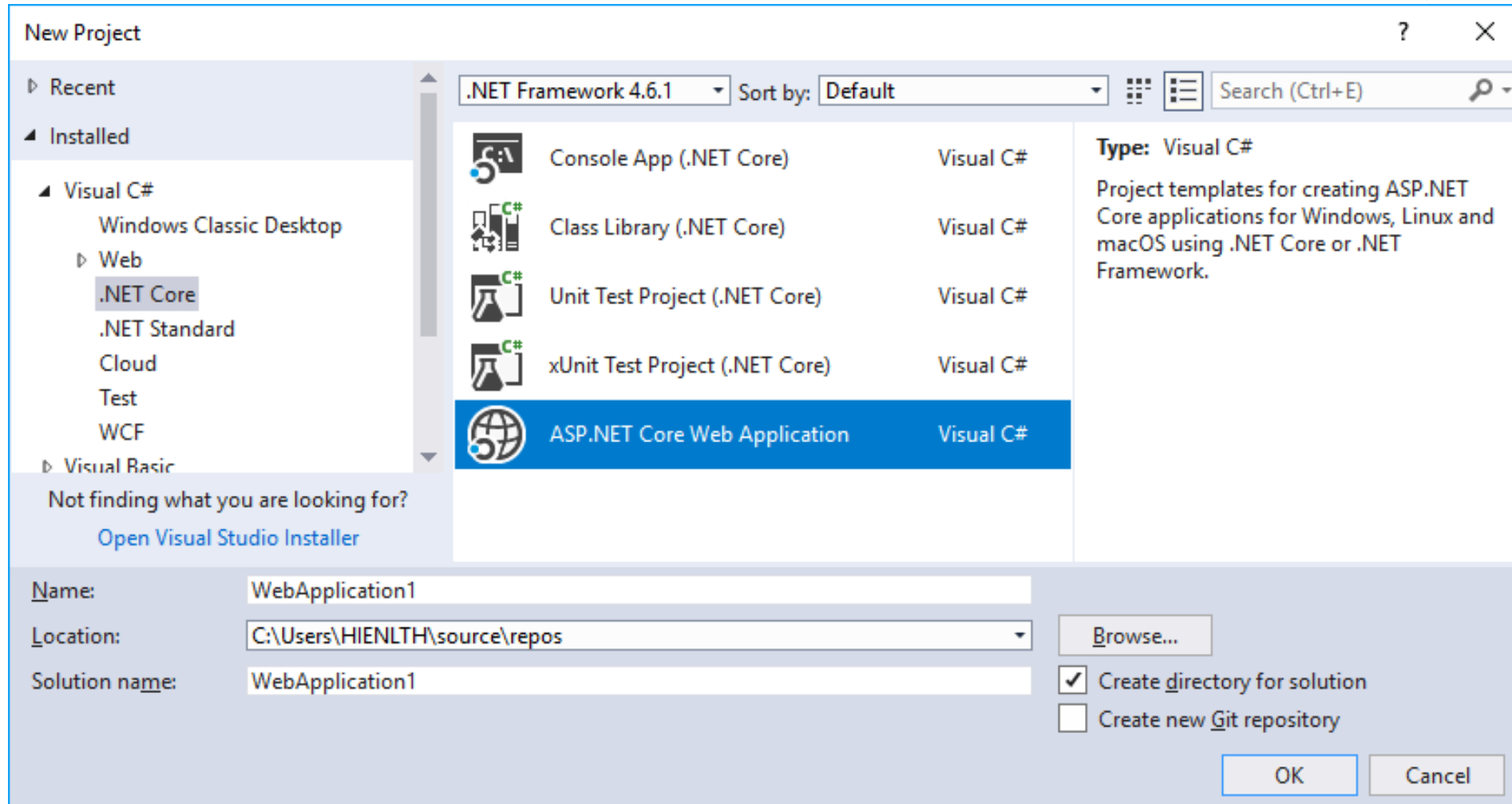
Select Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\HIENLTH> dotnet --version
2.1.4
PS C:\Users\HIENLTH> .
```

Visual Studio .NET Core Templates

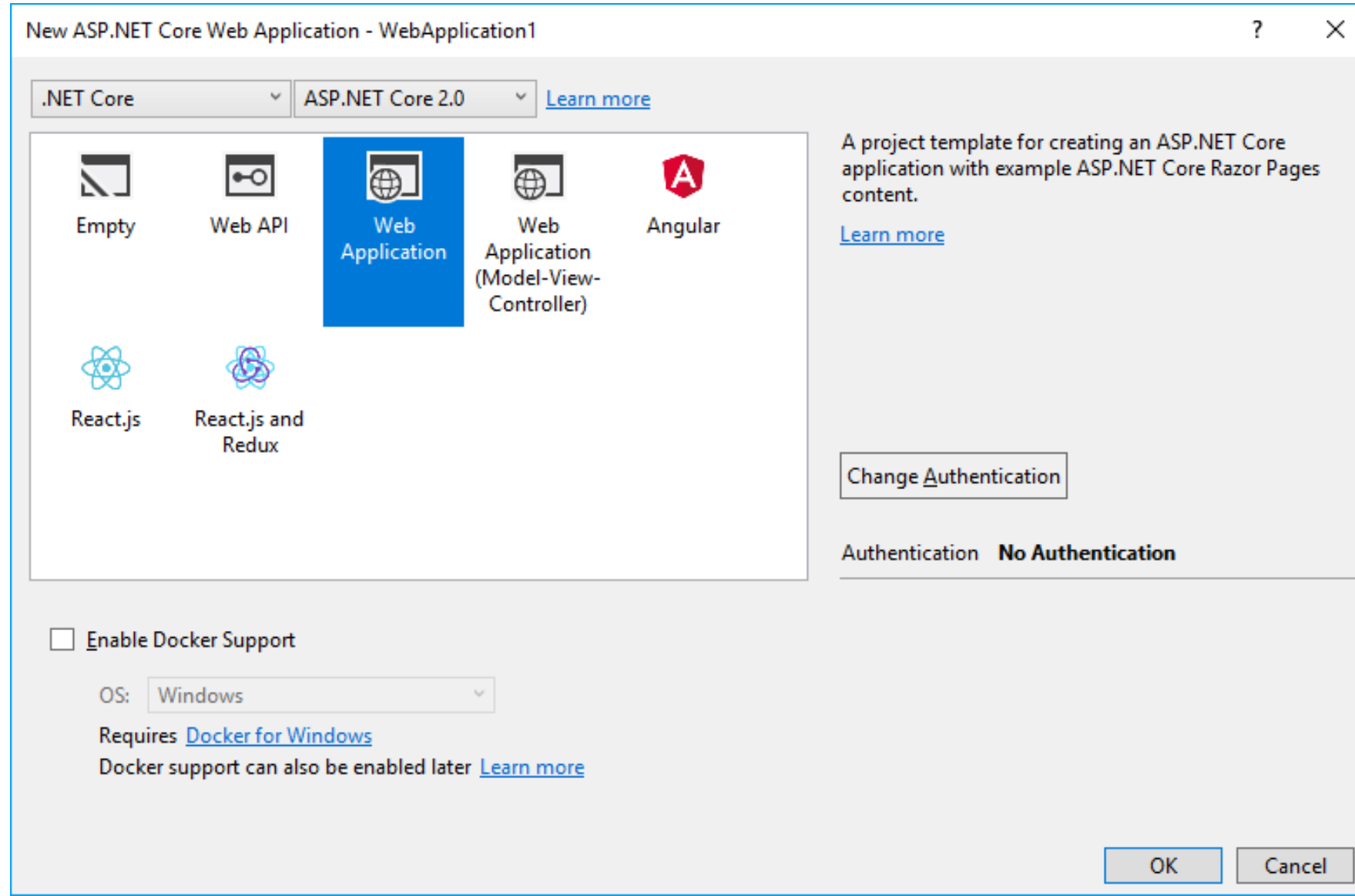
- ☐ Console App
- ☐ Class Library
- ☐ Unit Test
- ☐ xUnit Test
- ☐ ASP.NET Core



ASP.NET Core Project Templates

- ☐ Empty
- ☐ Web API
- ☐ Web App
- ☐ Web App (MVC)
- ☐ Angular
- ☐ React
- ☐ React and Redux

- ☐ Docker



New Web Projects with dotnet

❑ dotnet new web

- Empty Hello World Project

❑ dotnet new mvc

- MVC Template
- --auth Individual adds identity support
- --use-local-db true uses localdb instead of SQLite

❑ dotnet new webapi

❑ dotnet new razorpages

❑ dotnet new angular | react | reactredux

❑ Use --help to view options in CLI

dotnet CLI Templates

```
Windows PowerShell

Templates
```

| Templates | Short Name | Language | Tags |
|--|-------------|--------------|---------------------|
| Console Application | console | [C#], F#, VB | Common/Console |
| Class library | classlib | [C#], F#, VB | Common/Library |
| Unit Test Project | mstest | [C#], F#, VB | Test/MSTest |
| xUnit Test Project | xunit | [C#], F#, VB | Test/xUnit |
| ASP.NET Core Empty | web | [C#], F# | Web/Empty |
| ASP.NET Core Web App (Model-View-Controller) | mvc | [C#], F# | Web/MVC |
| ASP.NET Core Web App | razor | [C#] | Web/MVC/Razor Pages |
| ASP.NET Core with Angular | angular | [C#] | Web/MVC/SPA |
| ASP.NET Core with React.js | react | [C#] | Web/MVC/SPA |
| ASP.NET Core with React.js and Redux | reactredux | [C#] | Web/MVC/SPA |
| ASP.NET Core Web API | webapi | [C#], F# | Web/WebAPI |
| global.json file | globaljson | | Config |
| NuGet Config | nugetconfig | | Config |
| Web Config | webconfig | | Config |
| Solution File | sln | | Solution |
| Razor Page | page | | Web/ASP.NET |
| MVC ViewImports | viewimports | | Web/ASP.NET |
| MVC ViewStart | viewstart | | Web/ASP.NET |

```
Examples:
dotnet new mvc --auth Individual
dotnet new classlib --framework netcoreapp2.0
dotnet new --help
PS C:\Users\HIENLTH>
```

.NET Core 2.x CLI Commands

```
>dotnet --version
```

```
>dotnet --info
```

```
>dotnet new <TEMPLATE> --auth <AUTH_TYPE> -o <OUTPUT_DIR>
```

<TEMPLATE> = web | mvc | razor | angular | react | webapi

<AUTH_TYPE> for mvc,razor = None | Individual | SingleOrg | Windows

```
>dotnet new console -o MyConsoleApp
```

```
>dotnet new mvc --auth Individual -o MyMvcWebApp
```

```
>dotnet restore
```

```
>dotnet build
```

```
>dotnet run
```

<https://docs.microsoft.com/en-us/dotnet/core/tools/?tabs=netcore2x>

Azure CLI Commands

```
>az login  
>az group create -l <REGION> -n <RSG>  
>az appservice plan create -g <RSG> -n <ASP> --sku <PLAN> (e.g. F1)  
>az webapp create -g <RSG> -p <ASP> -n <APP>
```

RESULT → <http://<APP>.azurewebsites.net>

```
>git init  
>git add .  
>git commit -m "<COMMIT MESSAGE>"  
>az webapp deployment user set --user-name <USER>  
>az webapp deployment source config-local-git -g <RSG> -n <APP> --out tsv  
>git remote add azure <GIT URL>  
>git push azure master
```

GIT URL → <https://<USER>@<APP>.scm.azurewebsites.net/<APP>.git>

Demo Console App

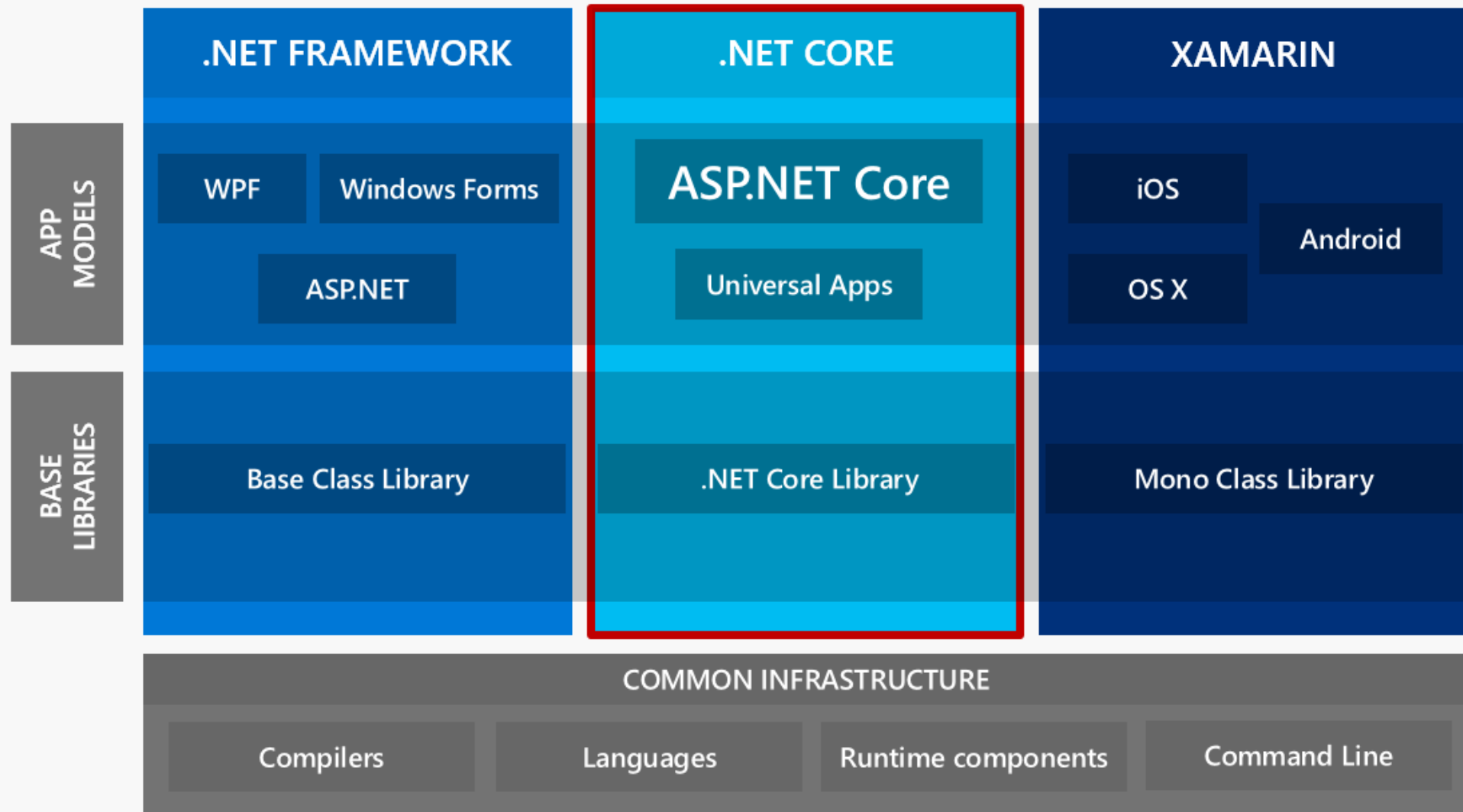
- ☐ Windows
- ☐ Linux (Ubuntu)

```
using System;

namespace ConsoleApplication
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Hello world!");
        }
    }
}
```


ASP.NET CORE

ASP.NET Core High-Level Overview



ASP.NET Core Features

Cloud-ready

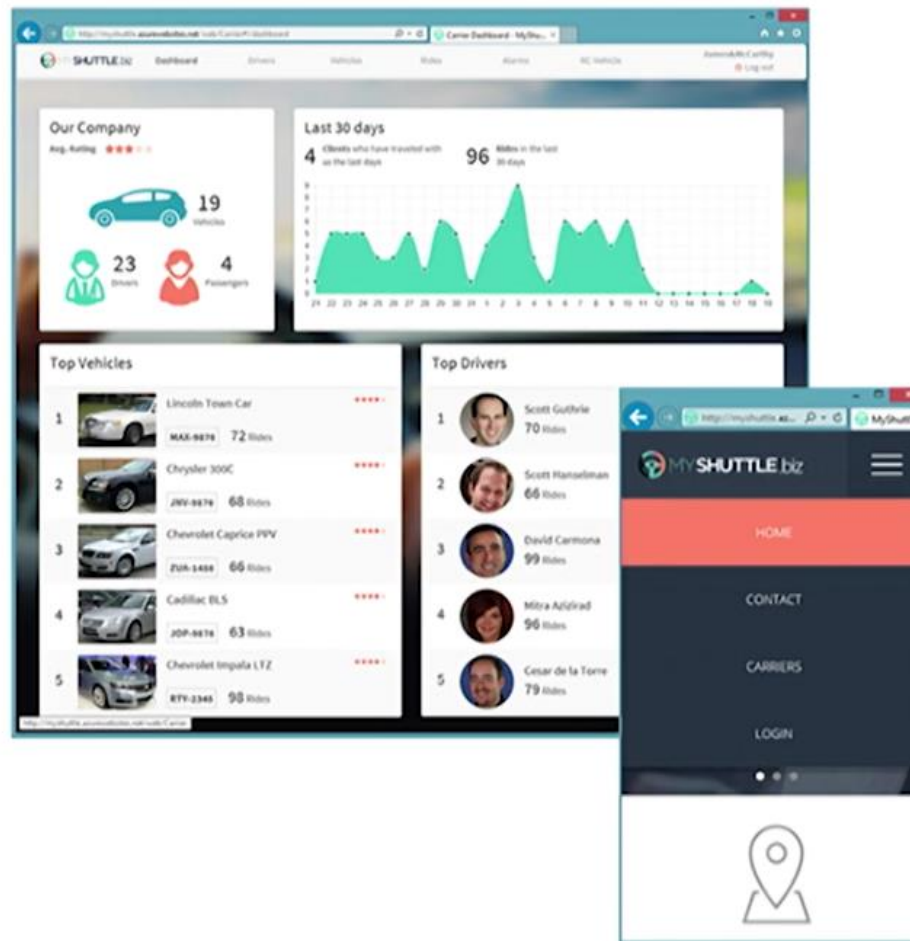
- Leaner, faster, simpler
- Designed from top to bottom to be ready for the cloud and cross-platform deployments

Modular and open

- More flexible with open source and modular implementation

Improved tooling and frameworks

- Deliver value faster with improved tooling and frameworks



NUMBER OF APIS

| Version | #APIs | Growth % |
|---------|--------|----------|
| 1.0 | 7,949 | |
| 1.1 | 10,239 | +29% |
| 1.2 | 10,285 | +0% |
| 1.3 | 13,122 | +28% |
| 1.4 | 13,140 | +0% |
| 1.5 | 13,355 | +2% |
| 1.6 | 13,501 | +1% |
| 2.0 | 32,638 | +142% |

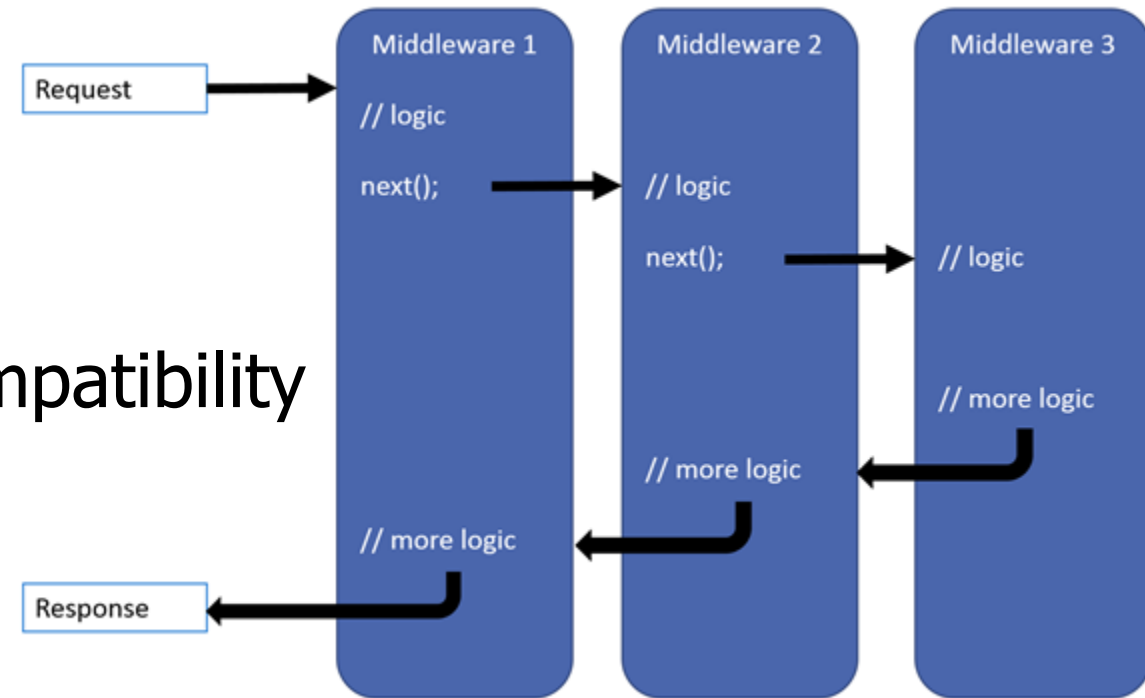
ASP.NET Core Summary

(MVC + Web API + Web Pages), SignalR

| Feature | Running on .NET 4.6 | Running on .NET Core |
|--|---------------------|----------------------|
| Cloud Ready | * | * |
| Modular Design | * | * |
| Dependency Injection | * | * |
| Consistent Tracing / Debugging | * | * |
| Faster Development (No Build Step) | * | * |
| Open Source | * | * |
| Full Side by Side (framework deployed inside application) | | * |
| Fast startup, Low memory / High throughput (best of class) | | * |

ASP.NET Core – what's different than ASP.NET 4

- ❑ ASP.NET Core is simply a console app.
- ❑ Web Forms is gone
- ❑ WCF server is gone
- ❑ Client side package management
- ❑ web.config is **only** around for IIS compatibility
- ❑ Global.asax is gone
- ❑ Startup.cs
- ❑ Middleware all the things, even MVC



Nice New Features over ASP.NET 4

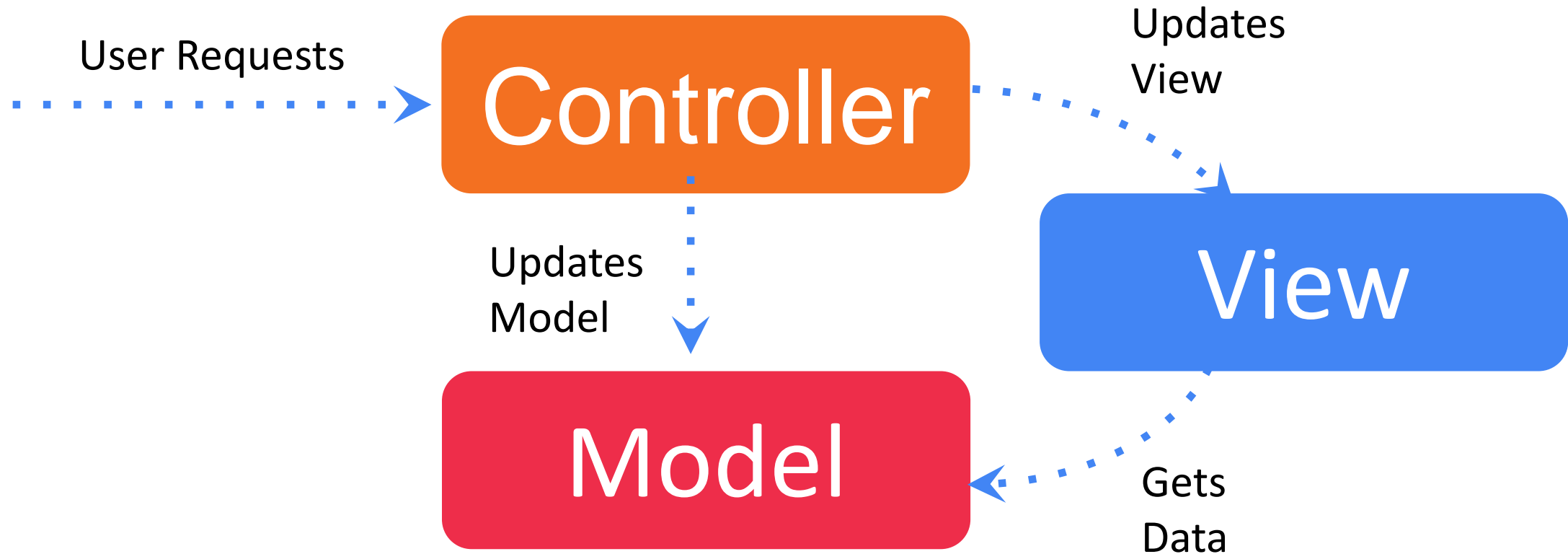
- ❑ Save and reload, no more building
- ❑ Dependency Injection built-in
- ❑ Environments are a first class citizen
- ❑ TagHelpers > Html Helpers

- HtmlHelper: `@Html.ActionLink("Home", "Index", "Home")`

- TagHelper: `Home`

ASP .NET CORE MVC

MVC Web App Basics



MVC (Web) Controllers

```
public class HumanController : Controller
{
    private readonly ApplicationDbContext _context;

    public HumanController(ApplicationDbContext context) {}

    // GET: Human, Human/Details/5
    public async Task<IActionResult> Index() {}
    public async Task<IActionResult> Details(int? id) {}

    // GET: Human/Create
    public IActionResult Create() {}

    // POST: Human
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Create([Bind("Id,FirstName,LastName")] Human human) {}

    // GET: Human/Edit/5
    public async Task<IActionResult> Edit(int? id) {}

    // POST: Human/Edit/5
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Edit(int id, [Bind("Id,FirstName,LastName")] Human human) {}
}
```



MVC (API) Controllers

```
public class ValuesController : Controller
{
    // GET: api/Values, api/Values/5
    [HttpGet]
    public IEnumerable<string> Get() {}

    [HttpGet("{id}", Name = "Get")]
    public string Get(int id) {}

    // POST: api/Values
    [HttpPost]
    public void Post([FromBody]string value) {}

    // PUT: api/Values/5
    [HttpPut("{id}")]
    public void Put(int id, [FromBody]string value) {}

    // DELETE: api/ApiWithActions/5
    [HttpDelete("{id}")]
    public void Delete(int id) {}
}
```

MVC (Web) Views

```
@model NiceStackWeb.Models.Human

@{
    ViewData["Title"] = "Details";
}

<h2>Details</h2>

<div>
    <h4>Human</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.FirstName)
        </dt>
        <dd>
            @Html.DisplayFor(model => model.FirstName)
        </dd> ...
    </dl>
</div>
<div>
    <a asp-action="Edit" asp-route-id="@Model.Id">Edit</a> |
    <a asp-action="Index">Back to List</a>
</div>
```

MVC Models

```
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;

public class Human
{
    [Key]
    public int Id { get; set; }

    [DisplayName("First Name")]
    public string FirstName { get; set; }

    [DisplayName("First Name")]
    public string LastName { get; set; }
}
```

New Razor Pages!

<http://www.hishambinateya.com/welcome-razor-pages>

The screenshot shows a web browser window with the address bar displaying 'hishambinateya.com/welcome-razor-pages'. The page content is as follows:

What are Razor Pages trying to do & What are not?

The razor pages are trying to:

- Make dynamic html and forms easier in ASP.NET Core.
- Reduce the number of folders and files in MVC structure that requires for page-focused scenarios.
- Simplify the code for page-focused pattern.
- Enable to return non html responses when necessary.
- Use MVC primitives as much as possible.
- Simplify the migration to traditional MVC structure.

The razor pages are not trying to:

- Create scripted page framework to compete with PHP .. etc.
- Hide C# with Domain Specification Language (DSL) in razor or otherwise.
- Create new primitives that only applicable for this model.

Intro to Razor Pages

The screenshot shows a web browser window displaying the Microsoft documentation page for Razor Pages. The browser's address bar shows the URL `docs.microsoft.com/en-us/aspnet/core/mvc/razor-pages`. The Microsoft logo and navigation menu are at the top. The page title is "Introduction to Razor Pages in ASP.NET Core", dated 2017-7-18, with a 16-minute read time. A left sidebar contains a table of contents with "Razor Pages" selected. The main content area lists topics covered in the article, such as prerequisites, project creation, and routing.

Introduction to Razor P. X +

docs.microsoft.com/en-us/aspnet/core/mvc/razor-pages

Microsoft Technologies Documentation Resources ASP.NET Core X Search

Docs Windows Microsoft Azure Visual Studio Office More

Docs / ASP.NET / ASP.NET Core

Comments Edit Share Theme Light

Filter

- Introduction
- Getting Started
- Razor Pages**
- Create a Web API
- Create a Web app
- Tutorials
- Fundamentals
- MVC
- Razor Pages**
- Razor syntax

Download PDF

Introduction to Razor Pages in ASP.NET Core

2017-7-18 • 16 min to read • Contributors

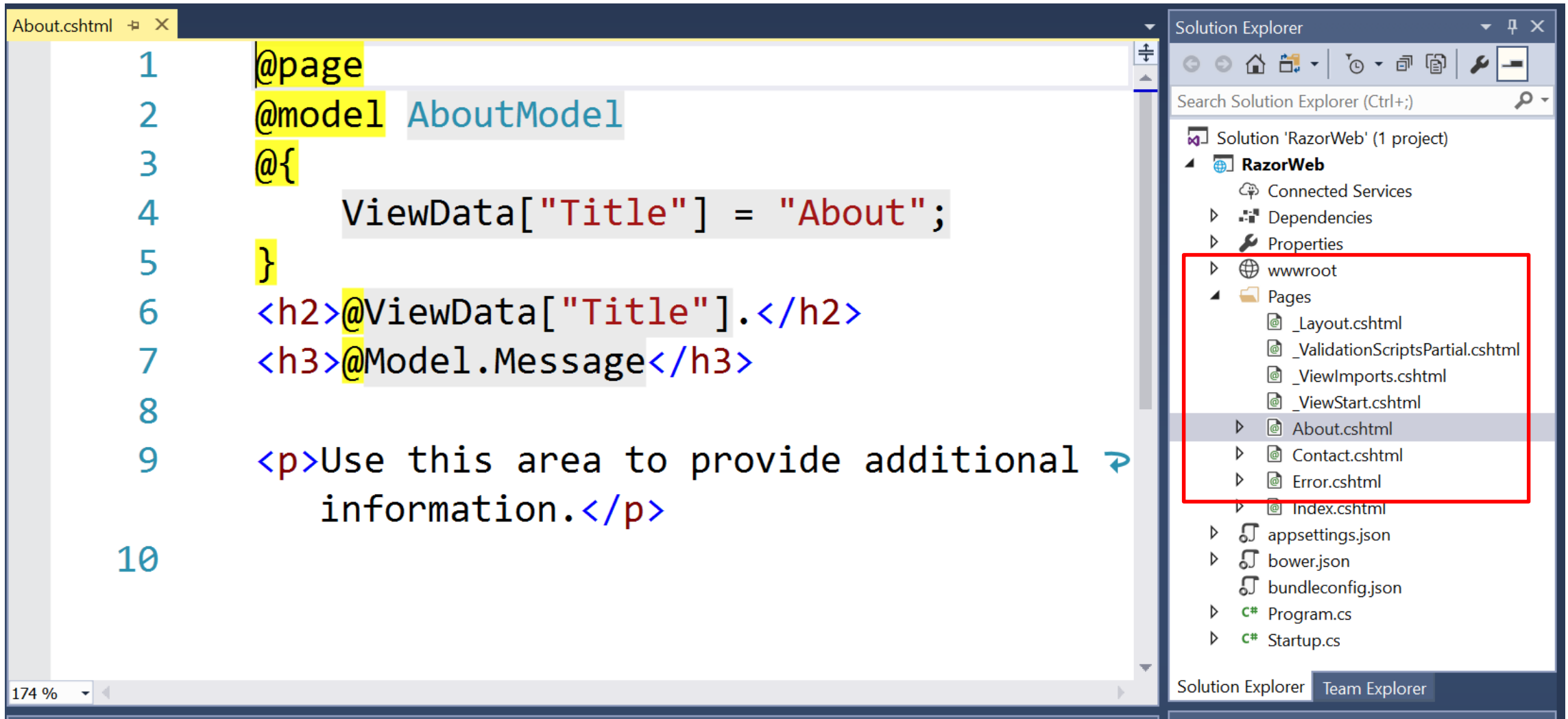
In this article

- ASP.NET Core 2.0 preview prerequisites
- Razor Pages
- Creating a Razor Pages project with Visual Studio 2017
- Writing a basic form
- XSRF/CSRF and Razor Pages
- Using Layouts, partials, templates, and Tag Helpers with Razor Pages
- URL generation for Pages
- TempData
- Multiple handlers per page
- Customizing Routing
- Configuration and settings

<https://docs.microsoft.com/visualstudio/products>

<https://docs.microsoft.com/en-us/aspnet/core/mvc/razor-pages>

Razor Syntax



The image shows a Visual Studio interface with a code editor on the left and a Solution Explorer on the right. The code editor displays the content of 'About.cshtml' with line numbers 1 through 10. The code uses Razor syntax, with tags like `@page`, `@model`, and `@{` highlighted in yellow. The Solution Explorer on the right shows the project structure for 'RazorWeb', with a red box highlighting the 'Pages' folder and its contents, including 'About.cshtml'.

```

1  @page
2  @model AboutModel
3  @{
4      ViewData["Title"] = "About";
5  }
6  <h2>@ViewData["Title"].</h2>
7  <h3>@Model.Message</h3>
8
9  <p>Use this area to provide additional
10     information.</p>

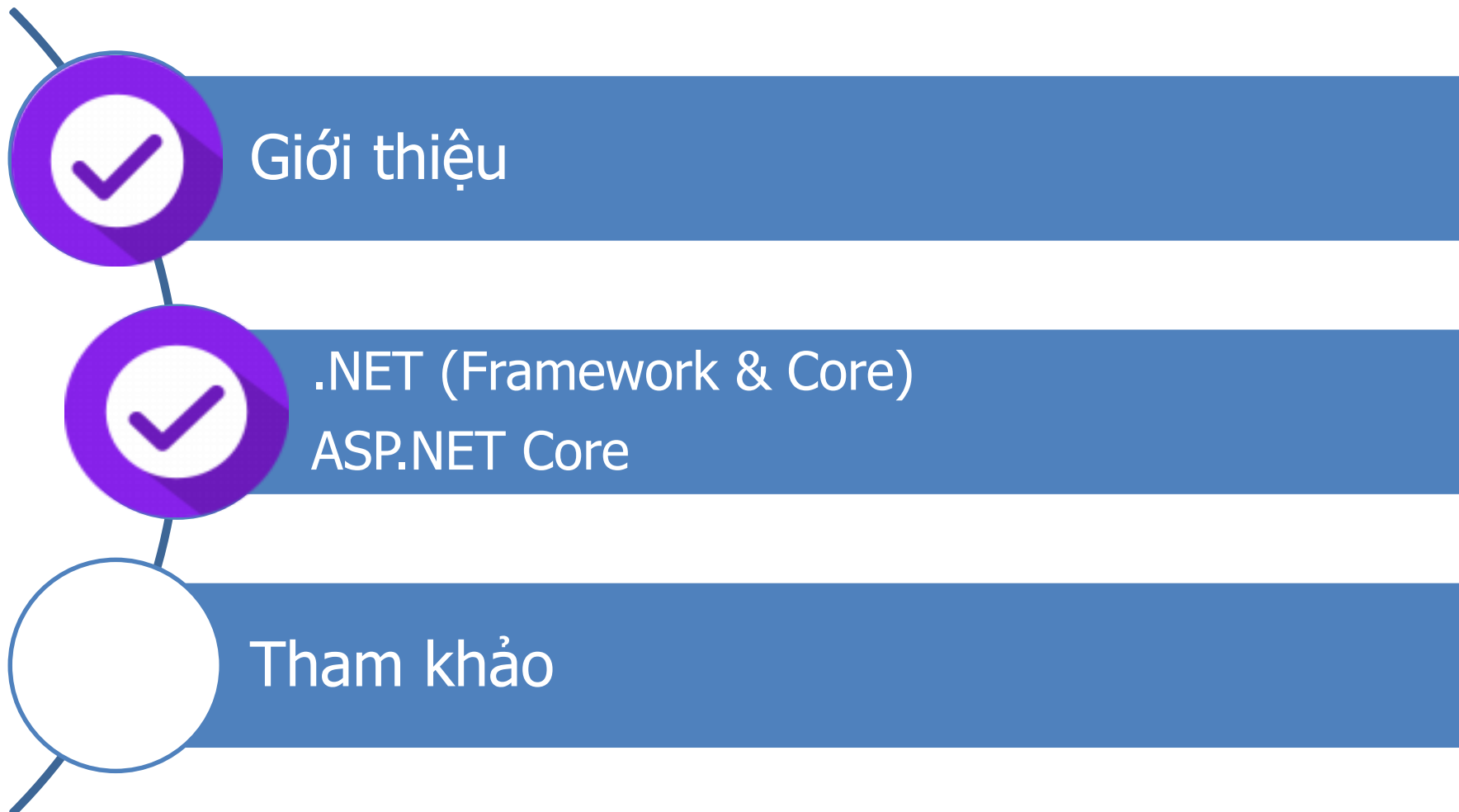
```

Solution Explorer:

- Solution 'RazorWeb' (1 project)
 - RazorWeb
 - Connected Services
 - Dependencies
 - Properties
 - wwwroot
 - Pages
 - _Layout.cshtml
 - _ValidationScriptsPartial.cshtml
 - _ViewImports.cshtml
 - _ViewStart.cshtml
 - About.cshtml
 - Contact.cshtml
 - Error.cshtml
 - Index.cshtml
 - appsettings.json
 - bower.json
 - bundleconfig.json
 - Program.cs
 - Startup.cs

Demo: Razor Pages

Demo: ASP.NET Core MVC



- ❑ ASP .NET: <http://www.asp.net>
- ❑ .NET Core: <https://www.microsoft.com/net>
- ❑ .NET Web Dev Blog: <https://blogs.msdn.microsoft.com/webdev>

- ❑ Tutorials: <https://docs.microsoft.com/en-us/aspnet/core>
- ❑ C# 7: <https://docs.microsoft.com/en-us/dotnet/articles/csharp/csharp-7>

- ❑ ASP.NET Core Roadmap: <https://github.com/aspnet/Home/wiki/Roadmap>
- ❑ .NET Core Roadmap: <https://github.com/dotnet/core/blob/master/roadmap.md>



Chương trình ASP.NET Core 2.0

❑ <http://nhatnghe.com/chuongtrinhhoc/aspcore>

❑ 108 giờ, 4 triệu

- Chiều T7, sang CN: 4 giờ/buổi
- Tối 357/246: 3 giờ/buổi

❑ Nội dung

- C# cơ bản
- HTML5/CSS3/jQuery/BootStrap cơ bản (FrontEnd development)
- ASP.NET Core MVC/Web App
- Entity Framework Core
- Web API

Resource for Seminar

□ <https://github.com/hienlth/aspcore>

THANKS FOR COMING!