Hướng dẫn tự học lập trình ASP.NET Core toàn tập

ASP.NET Core là một *framework* dành cho phát triển ứng dụng web mới nhất của Microsoft. Đây cũng là công nghệ Microsoft đầu tư phát triển nhiều nhất trong thời gian qua.

Trong thời gian qua, nhu cầu học và làm việc với ASP.NET Core đã tăng lên rất nhanh. Hiện nay đã có khá nhiều tài liệu bài bản và chuyên sâu về ASP.NET Core, từ sách in, sách điện tử, bài giảng video, đến các website/blog dành riêng cho chủ đề này.

Rất tiếc rằng các tài liệu đó phần lớn viết bằng tiếng Anh. Đây là một rào cản không nhỏ với các bạn sinh viên.

Để giúp các bạn có một tài liệu bài bản, hệ thống và chuyên sâu, Tự học ICT xây dựng tập bài giảng *Hướng dẫn tự học lập trình ASP.NET Core*. Tập bài giảng này sẽ đi từ những vấn đề cơ bản đến nâng cao để bạn có thể bắt nhịp với công nghệ mới này.

NỘI DUNG CỦA BÀI [Ẩn]

Giới thiệu chung về ASP.NET Core

ASP.NET Core là framework của Microsoft hỗ trợ phát triển các ứng dụng web. Phiên bản đầu tiên ASP.NET Core phát hành tháng 6 năm 2016.

Truyền thống và hiện đại

ASP.NET Core được thiết kế lại hoàn toàn để phù hợp cho phát triển các ứng dụng web hiện đại. Trong ASP.NET Core, cả framework và platform cho thực thi ứng dụng đều được xây dựng lại. Nó đồng thời bổ sung thêm những tính năng mới không có trong ASP.NET truyền thống.

Tuy được xây dựng lại từ đầu nhưng ASP.NET Core vẫn kế thừa những ưu điểm của ASP.NET (vốn đã và đang được sử dụng rất rộng rãi). Vì vậy, ASP.NET Core cũng được xem như người kế tục của ASP.NET. Những lập trình viên đã quen thuộc với ASP.NET rất dễ dàng chuyển sang ASP.NET Core.

Cộng đồng và doanh nghiệp

ASP.NET Core thu hút sự quan tâm rất lớn của cộng đồng do đây là một framework mã mở. Đây là điểm khác biệt rất lớn với ASP.NET truyền thống. Sự đóng góp của cộng đồng khiến ASP.NET Core phát triển rất nhanh chóng.

Đây là công nghệ mới nhất được Microsoft khuyến khích sử dụng để phát triển các ứng dụng web hiện đại, đặc biệt nếu đó là dự án mới. Các ứng dụng web sẵn có cũng có thể chuyển đổi sang ASP.NET Core để tận dụng những ưu thế của framework/platform mới này.

Hiện nay nhiều công ty đã và đang chuyển dịch sang ASP.NET Core, nhất là khi phát triển các dự án mới. Do vậy, nhu cầu học và làm việc với công nghệ mới này đang tăng lên rất nhanh.

Ưu nhược điểm của ASP.NET cũ

Để hiểu được vì sao Microsoft quyết định xây dựng một framework mới, cũng như thuyết phục bạn chuyển sang học ASP.NET Core, chúng ta sẽ nói qua một vài vấn đề của ASP.NET truyền thống.

Lịch sử của ASP.NET

ASP.NET ra đời từ 2002 với vai trò là một bộ phận của .NET Framework 1.0 nhằm thay thế cho ASP (Active Server Pages, ra đời từ 1996) cổ điển (sử dụng VBScript) và cạnh tranh với PHP.

Bên trên ASP.NET, Microsoft xây dựng một số mô hình lập trình khác nhau để hỗ trợ lập trình viên: (1) Web Forms, (2) MVC.

Web Forms ra đời năm 2002 hướng tới mô hình web "stateful" dựa trên sự kiện tương tự như Windows Forms. Mô hình lập trình của Web Forms có quá nhiều vấn đề, nhất là đối với các ứng dụng lớn, bao gồm hạn chế khi test, mô hình stateful quá phức tạp, không kiểm soát được HTML gây khó khăn cho xây dựng client. Web Forms dần bị thay thế khi MVC ra đời.

ASP.NET MVC ra đời năm 2009 dựa trên mô hình kiến trúc MVC (Model – View – Controller), tương tự như Ruby on Rails, Django hay Java Spring. Mô hình này đặc biệt thành công và được sử dụng rất rộng rãi thay thế cho Web Forms.

Năm 2012, ASP.NET Web API ra đời giúp phát triển ứng dụng dạng dịch vụ REST và dần thay thế WCF (Windows Communication Foundation, xuất hiện từ 2006).

Vấn đề của ASP.NET

Tất cả các mô hình lập trình của ASP.NET (Web Forms và MVC) đều được xây dựng trên cùng một framework chung sử dụng thư viện System.Web.dll – vốn là một bộ phận của .NET Framework lớn. Điều này mang tới cả ưu và nhược điểm.

Các ứng dụng web có thể sử dụng tất cả các tính năng của .NET Framework và ASP.NET chung – vốn vô cùng đa dạng, phong phú, có tính tin cậy và ổn định cao đã trải qua thử thách của thời gian.

Ở khía cạnh khác, ASP.NET chịu sự giới hạn về phát triển của .NET Framework nói chung (do là một bộ phận của .NET), ví dụ, về chu kỳ phát hành, vốn không phù hợp với các ứng dụng web hiện đại.

ASP.NET cũng gắn chặt với dịch vụ hosting của Windows sử dụng IIS (Internet Information Service). Điều này khiến ASP.NET không thể mở rộng ra hoạt động trên các hệ điều hành khác – vốn là một yêu cầu rất quan trọng hiện nay.

Những lý do trên dẫn tới việc Microsoft quyết định xây dựng một framework/platform hoàn toàn mới cho phát triển ứng dụng web và đặt tên nó là ASP.NET Core.

Như vậy, không nên nhầm lẫn giữa ASP.NET và ASP.NET Core. Chúng là hai framework hoàn toàn khác nhau.

.NET Core và ASP.NET Core

ASP.NET Core được thiết kế nhằm đáp ứng các yêu cầu:

- Phát triển và hoạt động đa nền tảng;
- Có kiến trúc dựa trên các module;
- Phát triển hoàn toàn ở dạng mã nguồn mở;
- Phù hợp với xu hướng hiện đại của ứng dụng web.

Để đạt được các yêu cầu trên, Microsoft xây dựng một platform mới đảm bảo nhẹ – nhanh – đa nền tảng. Platform mới này được đặt tên là **.NET Core**. Hiện nay .NET Core hoạt động được trên Windows, macOS và Linux.

> Như vậy .NET Core và .NET Framework sẵn có là hai platform độc lập, và bạn không nên nhầm lẫn giữa chúng.

Tuy nhiên, .NET Core chứa rất nhiều API giống như của .NET Framework. Điều này giúp lập trình viên dễ dàng chuyển đổi sang .NET Core mà không cần phải học lại mọi thứ từ đầu. Nếu nhìn từ phía các API thì có thể hình dung .NET Core tương tự như một bộ phận (độc lập) tách ra từ .NET Framework.

Ở những phiên bản đầu tiên .NET Core chỉ cho phép phát triển ứng dụng console đa nền tảng. ASP.NET Core là một tầng bổ sung xây dựng bên trên ứng dụng console để chuyển đổi nó thành một ứng dụng web.

> Từ phiên bản 3, .NET Core hỗ trợ thêm phát triển ứng dụng windows desktop, bao gồm windows forms và windows presentation foundation.

Cách nhìn nhận ứng dụng ASP.NET Core là một ứng dụng console có thể rất xa lạ với những bạn quen thuộc với ASP.NET. Tuy nhiên đây lại là một đặc điểm quan trọng của ASP.NET Core. Nó giúp ứng dụng viết trên ASP.NET Core có thể dễ dàng triển khai trên các hệ điều hành khác nhau, triển khai như một web server độc lập, hoặc kết hợp cùng các chương trình web server khác (IIS, Apache, Nginx).

> Bạn sẽ gặp lại vấn đề này trong một bài học riêng về mô hình hoạt động của ASP.NET Core.

ASP.NET Core và .NET Framework

Một điều khiến rất nhiều người nhầm lẫn là mối quan hệ giữa ASP.NET Core và .NET Framework: ASP.NET Core có thể hoạt động trên .NET Framework (giống như ASP.NET cũ).

Để hiểu vấn đề này bạn cần hình dung .NET Framework (và cả .NET Core) theo hai khía cạnh: (1) hệ thống thư viện API hỗ trợ phát triển ứng dụng; (2) runtime dành cho thực thi ứng dụng.

ASP.NET Core chứa hệ thống API của riêng nó. Hệ thống API này sử dụng các API cơ bản của .NET. Thêm vào đó, .NET Core và .NET Framework có chung hệ thống API cơ bản.

Runtime có thể hình dung như chương trình máy ảo sẽ nạp ứng dụng vào để thực thi. Ứng dụng và tất cả các thư viện của cả .NET Core và .NET Framework đều nằm ở dạng mã trung gian IL (Intermediate Language).

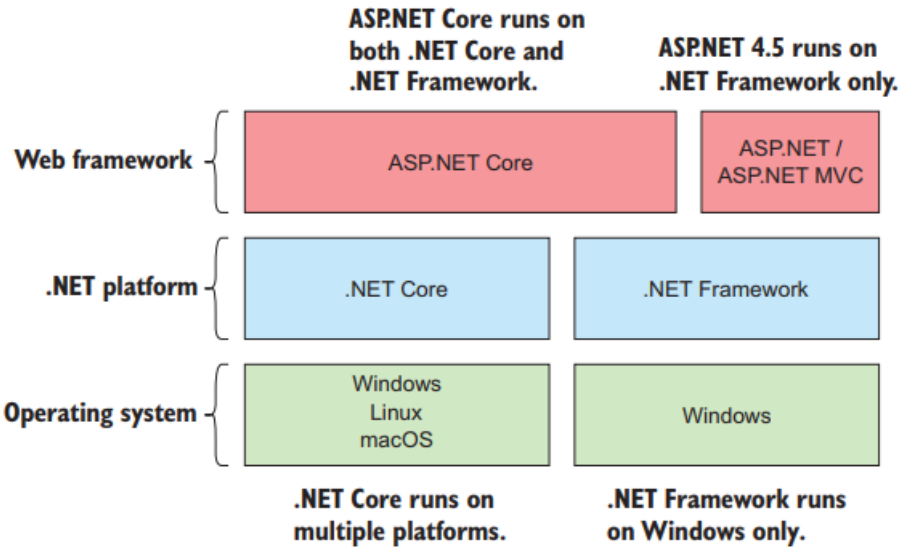
Hai yếu tố trên cho phép chương trình viết bằng ASP.NET Core có thể hoạt động trên runtime (máy ảo) của .NET Framework. Ở chiều ngược lại, ASP.NET truyền thống không thể hoạt động trên .NET Core do nó phụ thuộc vào System.Web.dll của .NET Framework, vốn không có trong .NET Core.

ASP.NET Core 2.0 tới 2.2 có thể chạy trên .NET Framework 4.6.1 (và các phiên bản cao hơn), đồng thời có thể chạy trên .NET Core 2.0 (và các bản cao hơn). Tuy nhiên ASP.NET Core 3.0 chỉ chạy trên .NET Core 3.0.

> Việc này có liên quan đến sự mở rộng của .NET Core so với .NET Framework. Để dễ hiểu hãy hình dung thế này:
> + Số lượng API của .NET Core 1.0 chỉ bằng một phần của .NET Framework 4.6.1 (phiên bản tương đương của .NET Framework tại thời điểm đó).
> + .NET Core 2.0 đạt lượng API tương đương với .NET Framework 4.7.1.
> + .NET Core 3.0 đã có số lượng API vượt quá của .NET Framework 4.8 – phiên bản cuối cùng của .NET Framework.

Khi chương trình ASP.NET Core thực thi trên runtime của .NET Framework, nó sẽ gắn chặt với Windows và IIS. Do đó nó mất đi những ưu thế của .NET Core. Bù lại, nó được hưởng lợi thế từ thư viện .NET Framework.

Mối quan hệ giữa ASP.NET Core với .NET Core và .NET Framework được minh họa như hình dưới đây.



Cũng theo sơ đồ này, ASP.NET truyền thống hoạt động hoàn toàn trên .NET Framework và gắn bó chặt chẽ với Windows và IIS.

ASP.NET và ASP.NET Core

Các phần trên đã giúp bạn hình dung ra vị trí và một số đặc điểm quan trọng của ASP.NET Core. Chúng ta sẽ nói tiếp về một số điểm khác biệt nữa giữa ASP.NET và ASP.NET Core từ khía cạnh lập trình ứng dụng.

Trong ASP.NET truyền thống bạn có thể xây dựng các ứng dụng web sử dụng một trong số các mô hình lập trình mà framework này cung cấp như Web Forms, MVC, Web API, Web Pages. Các mô hình này có thể xem như những framework riêng biệt xây dựng bên trên framework lớn ASP.NET.

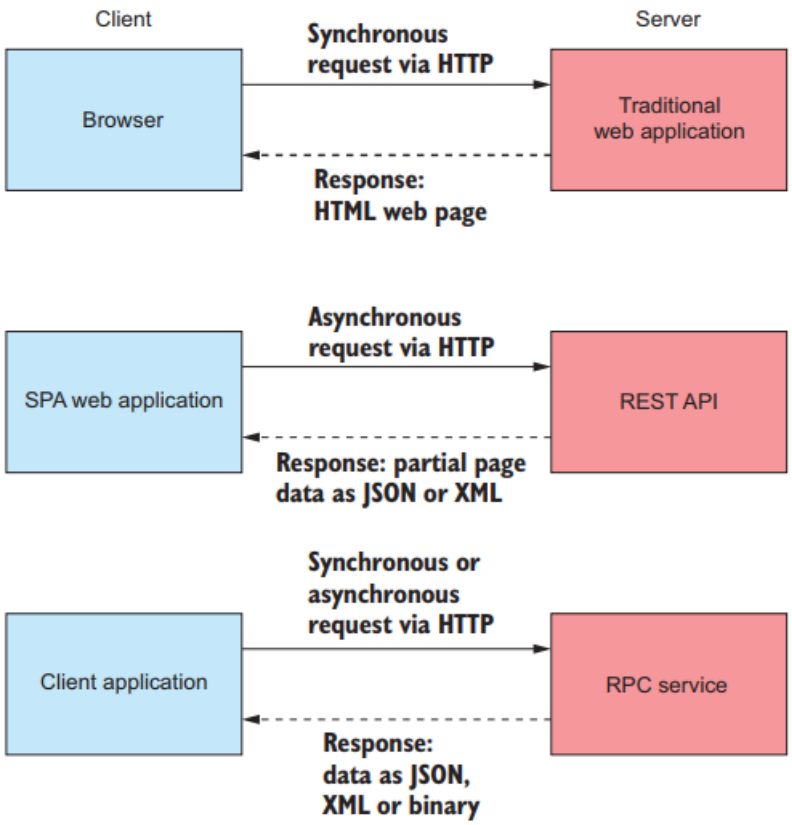
Trong ASP.NET Core không có sự phân chia như vậy nữa. Nói theo cách khác, các mô hình lập trình trong ASP.NET Core được thống nhất. Bạn có thể lựa chọn phát triển ứng dụng web theo mô hình MVC, Web API, Razor Pages. Tuy nhiên, các mô hình này không tách rời (sử dụng các class/thư viện riêng biệt) như trong ASP.NET mà nằm trong một hệ thống thống nhất, sử dụng chung class và thư viện.

> Tất cả những ưu điểm quen thuộc của MVC, Web API hay Web Pages của ASP.NET được kế thừa trong các mô hình tương đương của ASP.NET Core.

Bản thân các kỹ thuật lập trình của chúng hoàn toàn tương tự. Do đó, nếu bạn đã quen thuộc với ASP.NET, bạn rất dễ dàng tiếp cận với ASP.NET Core. Những khái niệm có sẵn trong ASP.NET như Router, Model-binding, Razor, v.v., có mặt đầy đủ trong ASP.NET Core.

Bạn có thể viết những loại ứng dụng gì trong ASP.NET Core

ASP.NET Core cho phép bạn viết gần như bất kỳ loại ứng nào có liên quan đến HTTP, như ứng dụng web HTML truyền thống, REST API cho ứng dụng đơn trang (Single Page Application, SPA), dịch vụ gọi hàm từ xa (Remote Procedure Call, RPC).



Ứng dụng web với mã HTML do server sinh ra là loại ứng dụng cơ bản và truyền thống mà ASP.NET Core hỗ trợ. Để phát triển các loại ứng dụng này bạn có thể sử dụng mô hình lập trình **MVC** hoặc **Razor Pages**.

ASP.NET Core MVC là mô hình lập trình ứng dụng web tương tự như ASP.NET MVC quen thuộc. Các thành phần của ứng dụng được phân chia ra các thành phần tuân theo mẫu kiến trúc MVC (Model – View – Controller), tương tự như Ruby on Rails, Java Spring hoặc Django.

Razor Pages là mô hình đơn giản hóa của MVC, chỉ bao gồm thành phần V (View) viết bằng ngôn ngữ Razor – loại cấu trúc kết hợp HTML và C#. Bạn có thể hình dung Razor là một dạng ngôn ngữ tương tự PHP nhưng có cấu trúc của C#. Razor Pages tương tự như Web Pages của ASP.NET.

Ứng dụng web đơn trang (**SPA**) hiện rất phổ biến. Loại ứng dụng này sử dụng các thư viện/framework JavaScript cho thành phần client. Các framework phổ biến hàng đầu hiện nay bao gồm Angular, React. Các loại ứng dụng này thường yêu cầu dữ liệu ở dạng JSON hoặc XML từ các REST (REpresentational State Transfer) API trên server.

Trên ASP.NET Core bạn có thể dễ dàng xây dựng các **REST API** như vậy. Bạn cũng có thể hình dung REST API này bao gồm thành phần M (Model) và C (Controller) của MVC nhưng không có thành phần V (View). Thành phần View do bản thân ứng dụng SPA chạy trên trình duyệt đảm nhiệm.

> Trên ASP.NET Core bạn cũng có thể sử dụng công nghệ phát triển ứng dụng SPA riêng mới nhất của Microsoft: **Blazor**. Hiện Blazor có hai mô hình, Blazor server và Blazor WebAssembly.

Gọi hàm từ xa (**RPC**) cũng là một mô hình sử dụng trong phát triển ứng dụng hướng dịch vụ (Service-Oriented Application). Trước đây WCF (Windows Communications Foundation) là công cụ chủ yếu phục vụ cho mục đích này. Tuy nhiên WCF đã dừng phát triển. Bạn có thể sử dụng ASP.NET Core thay thế cho mục đích này.

Nếu không muốn tự mình xây dựng mọi thứ từ đầu, bạn cũng có thể sử dụng một hệ quản trị nội dung (Content Management System, **CMS**) xây dựng trên ASP.NET Core như Orchard Core hoặc Piranha. Các CMS giúp bạn nhanh chóng xây dựng ra các website với các tính năng cơ bản. Bạn chỉ cần phát triển những gì mình cần mà không phải xây dựng mọi thứ từ đầu.

Lộ trình học ASP.NET Core

Ở trên bạn đã thấy ASP.NET Core hỗ trợ phát triển nhiều loại ứng dụng khác nhau với các mô hình lập trình riêng biệt: Razor pages, MVC, Blazor, React/Angular, Web API, SignalR, gRPC. Sự đa dạng này làm những người mới bắt đầu tiếp xúc với ASP.NET Core lúng túng không biết nên bắt đầu từ đâu.

Nếu bạn có xuất phát điểm là PHP, ASP cổ điển, ASP.NET Web Forms, hoặc ASP.NET Web Pages, bạn nên bắt đầu với Razor Pages. Sau đó, bạn có thể tiếp tục với ASP.NET Core MVC. Đây cũng là lộ trình được Microsoft khuyến nghị nếu bạn là người mới học phát triển ứng dụng web nói chung.

Nếu bạn đã quen thuộc với ASP.NET MVC hoặc ASP.NET Web API, bạn nên bắt đầu với ASP.NET Core MVC vì chúng hoàn toàn tương đồng và bạn sẽ rất nhanh chóng nắm bắt được.

Khi đã học xong ASP.NET Core MVC, bạn nên làm quen với một hệ quản trị nội dung (Content Management System, CMS) nào đó. CMS sẽ giúp bạn rất nhiều nếu cần xây dựng các ứng dụng nhanh. Một số CMS trên .NET Core thường dùng là Piranha và Orchard Core.

Nếu muốn đi theo hướng phát triển web client (chương trình chạy trên trình duyệt), bạn nên học tiếp SignalR và Blazor.

Kết luận

Bài học này chỉ mang tính chất giới thiệu chung để bạn có được hình dung đại thể về đặc điểm và vị trí của ASP.NET Core. Cụ thể, chúng ta đã điểm qua khái niệm và một số đặc

điểm quan trọng của ASP.NET Core, mối quan hệ giữa ASP.NET Core với .NET Core, .NET Framework và ASP.NET truyền thống.

> + Nếu bạn thấy site hữu ích, trước khi rời đi hãy **giúp đỡ** site bằng một hành động nhỏ để site có thể phát triển và phục vụ bạn tốt hơn.
> + Nếu bạn thấy bài viết hữu ích, hãy giúp **chia sẻ** tới mọi người.
> + Nếu có thắc mắc hoặc cần trao đổi thêm, mời bạn viết trong phần **thảo luận** cuối trang.
> Cảm ơn bạn!

Chúc bạn học tốt!

Tutorial Content

Thực hành (2) CRUD trong ASP.NET Core MVC

- Thực hành (3) tìm kiếm, sắp xếp, phân trang, upload, download

ASP.NET CORE WEB API

- Web API trong Asp.net Core

MỘT SỐ CHỦ ĐỀ NÂNG CAO

- Partial Page trong ASP.NET Core
- View component trong ASP.NET Core
- Tag Helper trong ASP.NET Core
- Sử dụng tag helper xây dựng sẵn của ASP.NET Core
- Thẩm định dữ liệu (model validation) trong ASP.NET Core
- Kiến trúc ứng dụng web trong ASP.NET Core

ASP.NET CORE BLAZOR

- Blazor – client web app với C# (và không JavaScript)
- Mô hình hoạt động của Blazor Server
- Mô hình hoạt động của Blazor WebAssembly
- Xây dựng ứng dụng Blazor server từ A-Z (+video)
- Mô hình code-behind trong Blazor