

Bài: Giới thiệu tổng quan công nghệ Web ASP.Net MVC

Xem bài học trên website để ủng hộ Kteam: [Giới thiệu tổng quan công nghệ Web ASP.Net MVC](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Những ai yêu thích lập trình web sử dụng công nghệ **ASP.NET** của Microsoft thì chắc chắn một điều là ai cũng phải biết đến Webform, một công nghệ khá cũ của Microsoft giúp lập trình web. Tuy nhiên với công nghệ ASP.Net webform thì có rất nhiều nhược điểm và một trong nhược điểm đáng chú ý là giao diện webform phải sử dụng toolbox sẵn có, do đó chúng ta mất linh hoạt trong việc điều khiển giao diện.

Vì vậy, “ông lớn” Microsoft đã cho ra đời 1 công nghệ lập trình web mới đó là **ASP.Net MVC** với rất nhiều ưu điểm. Vậy những ưu điểm này là gì và sức mạnh của công nghệ này như thế nào?

Theo dõi tiếp nhé, mình sẽ giới thiệu ngay sau đây!

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- Kiến thức căn bản và hướng đối tượng C#
- Kiến thức về **ASP.Net webform** là 1 lợi thế

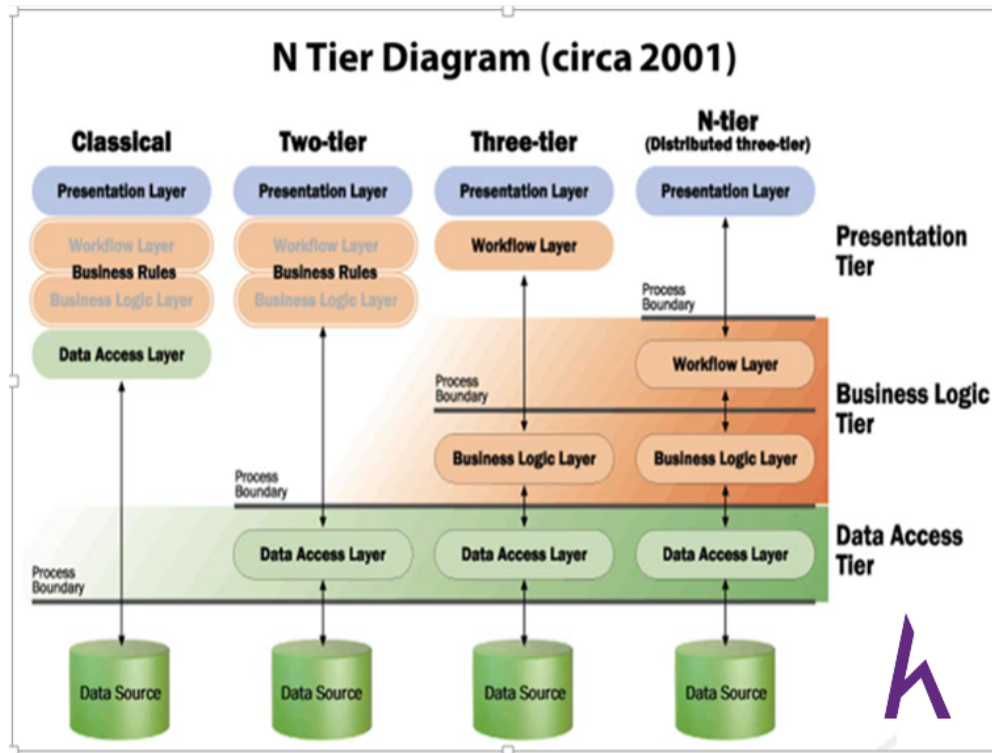
Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Tổng quan về mô hình MVC
- Tổng quan về mô hình lập trình web **ASP.Net MVC**

Tổng quan mô hình MVC

Trước khi đi vào tìm hiểu về mô hình MVC thì mình nhắc lại cho các bạn về cấu trúc cơ bản của mô hình lập trình web hiện nay

- Minh họa cấu trúc lập trình web:



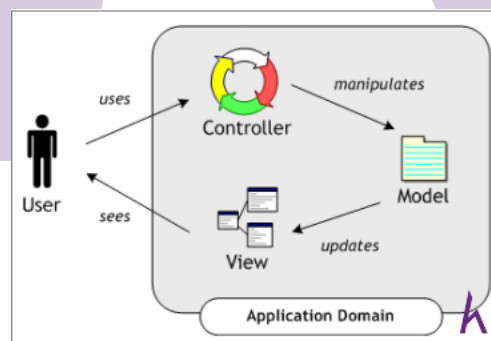
Trong lập trình web chúng ta chia ra 3 tầng như sau:

1. **Presentation Layer** (tầng 3): tầng này giúp hiển thị giao diện
2. **Business Logic Layer** (tầng 2): tầng này chứa các hàm thực thi, giúp xử lý sự kiện xảy ra trên tầng Presentation Layer.
3. **Data Access Layer** (tầng 1): tầng này chứa hàm kết nối với các hệ quản trị cơ sở dữ liệu như SQL Server, Mysql, Oracle ...

- Theo như hình mô tả, cấu trúc cổ điển là 3 tầng này trộn vào nhau hay còn gọi là cấu trúc **One - Tier**. Khi làm việc với cấu trúc **One - Tier** chúng ta gặp khó khăn trong việc chỉnh sửa code hay chỉnh sửa giao diện vì các tầng này bị trộn vào nhau.
- Tiếp đó, cấu trúc **Two - Tier** ra đời, tuy nhiên vẫn chưa giải quyết vấn đề của cấu trúc **One - Tier** gặp phải đó là việc chỉnh sửa code hay giao diện gặp khó khăn vì tầng 2 và tầng 3 vẫn chưa tách biệt khỏi nhau.
- Vì vậy cấu trúc **Three - Tier** đã ra đời và giải quyết tất cả các vấn đề mà **One - Tier**, **Two - Tier** gặp phải. Lúc này 3 tầng đã tách biệt nhau ra do đó việc chỉnh sửa code hay giao diện sẽ trở nên dễ dàng hơn giúp cho ứng dụng tối ưu hóa nhất

Sau khi hiểu và nhớ lại cấu trúc của một mô hình lập trình web cơ bản thì chúng ta sẽ đi qua tìm hiểu về mô hình MVC

- Minh họa mô hình MVC:



MVC (**M**odel - **V**iew - **C**ontroller) là một design pattern đã tồn tại rất lâu trong ngành công nghệ phần mềm. Một ứng dụng viết theo mô hình MVC sẽ bao gồm 3 thành phần tách biệt nhau đó là Model, View, Controller. Giống như trong cấu trúc **Three - Tier**, mô hình MVC giúp tách biệt 3 tầng trong mô hình lập trình web, vì vậy giúp tối ưu ứng dụng, dễ dàng thêm mới và chỉnh sửa code hoặc giao diện

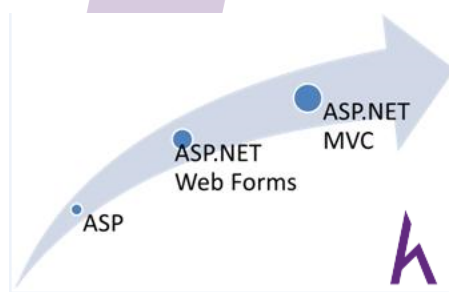
- **Model**: ở phần trước mình đã nhắc lại cho các bạn về 3 tầng trong mô hình
 - Three – Tier thì trong đó gồm có 2 tầng **Data Access Layer** và tầng **Business Logic Layer**. Hai tầng này là hai tầng tương đương với tầng model trong mô hình MVC.
- **View**: là tầng giao diện, hiển thị dữ liệu được truy xuất từ tầng model. Tầng này tương đương với tầng **Presentation Layer** trong cấu trúc Three – Tier.
- **Controller**: đây là tầng giúp kết nối giữa tầng model và tầng view trong mô hình MVC, có nghĩa là nếu phía client yêu cầu hiển thị dữ liệu thì controller gọi giữ liệu từ model và trả về cho view vì view tương tác trực tiếp với client

Ví dụ: User yêu cầu hiển thị thông tin cá nhân của user

1. User gửi một yêu cầu tới **controller**.
2. **Controller** nhận yêu cầu, xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì **controller** sẽ gửi yêu cầu đó xuống tầng **model** để truy xuất dữ liệu
3. Tầng **model** sẽ lấy dữ liệu từ database sau đó truyền dữ liệu qua tầng view thông qua tầng **controller** để tầng view hiển thị dữ liệu cho User
4. User sẽ thấy thông tin hiển thị ở giao diện và cụ thể ở đây là tầng view

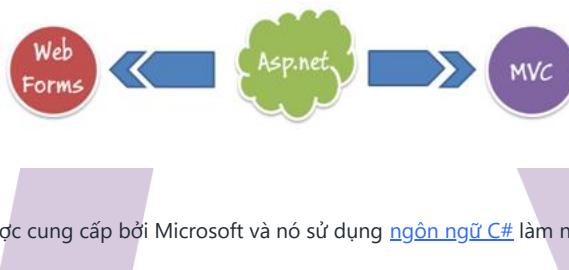
Qua ví dụ minh họa thì các bạn đã hiểu được cơ chế hoạt động của mô hình MVC rồi! Bây giờ mình sẽ đi qua tổng quan về **ASP.Net MVC**

Tổng quan về mô hình lập trình web ASP.Net MVC



Hình mô tả phía trên cho các bạn thấy trước khi **ASP.Net MVC** ra đời thì **ASP.Net webform** là 1 framework lập trình web khá quen thuộc đối với những tín đồ của **ASP.Net**. Tuy nhiên, khi **ASP.Net MVC** xuất hiện thì dường như **ASP.Net webform** bị lãng quên. Để hiểu được lý do tạo sao lại như vậy thì chúng ta sẽ đi từng bước tìm hiểu nhé!

ASP.Net là gì



ASP.NET là 1 framwork lập trình web được cung cấp bởi Microsoft và nó sử dụng [ngôn ngữ C#](#) làm ngôn ngữ phát triển. **ASP.NET** bao gồm 2 model đó là **Web Forms** và **MVC**.

Nhược điểm khi lập trình với ASP.Net web forms:

Trong nội dung bài ngày hôm nay, mình tập trung giới thiệu **ASP.Net MVC** nên mình sẽ không đi vào cụ thể về **Web Forms** mà mình chỉ đưa ra 1 vài nhược điểm của **Web Forms**.

- **Web Forms** không có sự tách biệt rõ ràng 3 tầng trong mô hình lập trình web.
- Trong **Web Forms** có sử dụng 1 đối tượng View State mà đối tượng này làm cho trang web bị tăng kích thước vì vậy làm giảm hiệu năng của ứng dụng.

- Giao diện của **Web Forms** được thiết kế dựa trên những toolbox có sẵn mà web forms cung cấp mà hiện nay đa phần designer sử dụng html, css để thiết kế website vì nó gây khó khăn trong việc thiết kế web hiện nay, việc thiết kế trở nên mất linh hoạt.

Với những khó khăn như vậy thì Microsoft đã cho ra đời **ASP.Net MVC**.

Cơ chế hoạt động và những ưu điểm trong lập trình web với ASP.Net MVC

ASP.Net MVC là 1 framework lập trình web mới của microsoft, công nghệ này ứng dụng mô hình MVC vào trong ASP.Net, còn mô hình MVC là như thế nào thì mình đã nói rõ ở phần phía trên rồi nhé

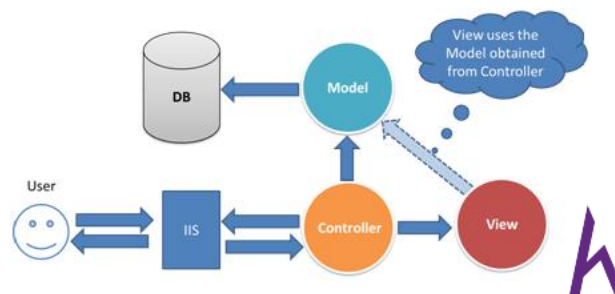
Stable Release	.NET Framework	Included in Visual Studio IDE	Year
ASP.NET MVC 1.0	.Net Framework 3.5	Visual Studio 2008, 2008 SP1	13-Mar-2009
ASP.NET MVC 2	.Net Framework 3.5, 4.0	Visual Studio 2008, 2010	10-Mar-2010
ASP.NET MVC 3	.Net Framework 4.0	Visual Studio 2010	13-Jan-2011
ASP.NET MVC 4	.Net Framework 4.0, 4.5	Visual Studio 2010 SP1, 2012	15-Aug-2012
ASP.NET MVC 5	.Net Framework 4.5, 4.5.1	Visual Studio 2013, Community	17-Oct-2013

Qua hình mô tả phía trên cho các bạn thấy tuy **ASP.Net MVC** ra đời năm 2009 nhưng mà tính tới năm 2013 thì nó đã được update lên tới phiên bản **ASP.Net MVC 5**, đây cũng là 1 trong những ưu điểm của ông lớn Microsoft cập nhật công nghệ liên tục.

Và mới đây thì **ASP.Net** cũng đã cho ra đời 1 công nghệ mới đó là **ASP.Net core 1.0**

Với sự cập nhật thường xuyên như vậy đã giúp cho **ASP.Net MVC** ngày nay phát triển mạnh mẽ không thua kém gì các công nghệ lập trình web sử dụng Java và Php.

Cơ chế hoạt động của ASP.Net MVC



Dựa vào hình mô phỏng phía trên thì mình sẽ khái quát sơ qua cơ chế hoạt động của **ASP.Net MVC**

- User gửi 1 yêu cầu tới server bằng cách truyền vào 1 URL trong browser
- Yêu cầu đó được gửi tới controller đầu tiên, controller sẽ xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì controller sẽ chuyển qua tầng model
- Tại tầng model, dữ liệu được truy xuất từ database và sau đó truyền qua view thông qua controller
- Controller sẽ giúp dữ liệu được chuyển từ model qua view
- View là tầng cuối cùng giao tiếp với User, mọi dữ liệu sẽ được hiển thị cho User thông qua tầng View

Những ưu điểm mà ASP.Net MVC mang lại cho chúng ta:

- Do sử dụng mô hình MVC nên trong **ASP.Net MVC** đã tách biệt được các tầng trong mô hình lập trình web vì vậy giúp tối ưu ứng dụng và dễ dàng trong việc viết code, giao diện
- Giao diện trong **ASP.Net MVC** sử dụng công nghệ thiết kế web HTML, CSS nên việc thiết kế giao diện trở nên dễ dàng và giúp cho designer linh hoạt trong việc thiết kế

3. ASP.Net MVC không sử dụng view state vì vậy trang web không bị tăng kích thước do đó hiệu năng hoạt động không bị giảm

ASP.Net MVC đã khắc phục được các nhược điểm của web forms vì vậy web forms hiện nay không còn được dùng phổ biến nữa.

Kết luận

Qua bài học ngày hôm nay, mình đã giúp các bạn hiểu khái quát về công nghệ lập trình web **ASP.Net MVC** của Microsoft.

Bài sau chúng ta sẽ đi vào [Tạo 1 project ASP.Net MVC đơn giản và giải thích cấu trúc, vai trò của View, Controller trong mô hình MVC. Xuất câu "Hello Howkteam!"](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".

