### [Project MVC toàn tập - Mô hình MVC là gì? - freetuts](http://freetuts.net/mvc-php-mo-hinh-mvc-la-gi-354.html)

freetuts.net › PHP › MVC PHP nâng cao

31 thg 3, 2015 - **Trong PHP** hiện tại có khá nhiều Framework và tất cả đều xây dựng từ mô hình **MVC**, từ đó bạn có thể thấy sự quan **trọng** của nó như thế nào ...

### [Mô hình MVC trong PHP, tìm hiểu mô hình MVC PHP thông qua ví dụ](http://kungfuphp.com/php-nang-cao/tim-hieu-ve-mo-hinh-mvc-trong-php.html)

kungfuphp.com/php-nang-cao/tim-hieu-ve-mo-hinh-mvc-trong-php.html

3 thg 2, 2015 - Hướng dẫn chi tiết để bạn hiểu về mô hình **MVC trong PHP**, cách hoạt động của **MVC trong PHP** như thế nào thông qua bài thực hành ví dụ cụ ...

### [PHP Framework - Mô hình MVC trong php - Học lập trình PHP](http://phpandmysql.net/php-framework/mo-hinh-mvc-trong-php-47.html)

phpandmysql.net/php-framework/mo-hinh-mvc-trong-php-47.html

Đôi lúc bạn nghe những khái niệm nào đó thì lại thấy lạ lẫm. Nhưng thật sự những khái đó luôn đi từ những luận điểm căn bản mà có thể bạn đã biết. Cái bạn ...

### [Mô Hình MVC trong php - hoc php nang cao - hocphp.info](http://hocphp.info/mo-hinh-mvc/)

hocphp.info › Php nâng cao

4 thg 9, 2014 - **MVC** là mô hình quen thuộc với các bạn đã học framework hay đã tìm ... Các bạn tạo class Bootstrap **trong** file Bootstrap.**php** với nội dung sau: ...

### [Khái niệm Mô hình MVC trong PHP - Thiết kế web](http://thietkeweb9999.com/blogs/thiet-ke-website/1476-khai-niem-mo-hinh-mvc-trong-php)

thietkeweb9999.com/blogs/thiet-ke.../1476-khai-niem-mo-hinh-mvc-trong-php

Khái niệm mô hình **MVC**. **Trong** bài học trước, chúng tôi đã giới thiệu **PHP** Framework CodeIgniter, giờ chúng tôi sẽ giới thiệu tiếp cho bạn về mô hình VMC.

### [MVC Là Gì? Mô hình MVC Trong PHP Cơ Bản Và PHP Nâng Cao](https://vietpro.net.vn/php-mvc)

https://vietpro.net.vn/php-mvc

Xếp hạng: 3,7 - ‎12 phiếu bầu

Giới thiệu về Mô hình **MVC trong PHP** cơ bản. Xin chào các bạn, ở chuyên mục này mình xin phép được giới thiệu sơ lược về mô hình MVC để các bạn có thể ...

### [Xây dựng website với mô hình MVC trong PHP - YouTube](https://www.youtube.com/watch?v=JfbLk2OqJCM)

[**▶ 49:46**](https://www.youtube.com/watch?v=JfbLk2OqJCM)

https://www.youtube.com/watch?v=JfbLk2OqJCM

13 thg 3, 2015 - Tải lên bởi Tran manh thang na

Series xây dựng website với mô hình **MVC trong PHP**. Đầy đủ về php ở đây các bạn nhé: http://thangtm.info/tag/hoc-lap-trinh ...

### [Mô hình MVC cơ bản trong PHP - Học lập trình online miễn phí](https://www.google.com.vn/interstitial?url=http://tailieulaptrinh.com/php/mo-hinh-mvc-co-ban-trong-php.html)

tailieulaptrinh.com/php/mo-hinh-mvc-co-ban-trong-php.html

[Trang web này có thể gây hại cho máy tính của bạn.](https://www.google.com.vn/support/answer/45449?topic=360&hl=vi)

30 thg 12, 2015 - với các bạn mới tìm hiểu **PHP** thì mô hình **MVC** vẫn còn hơi khó hiểu, do đó bài viết này sẽ nói cụ thể về mô hình **MVC** cơ bản **trong PHP** và ...

### [Code mô hình MVC PHP đơn giản - SinhvienIT](http://sinhvienit.net/forum/code-mo-hinh-mvc-php-don-gian.315101.html)

sinhvienit.net › Diễn đàn › Ngành công nghệ phần mềm › Lập trình web › PHP

14 thg 4, 2014 - Mô hình **MVC** được sử dụng **trong** nhiều **PHP** framework nổi tiếng như Zend Framework, CakePHP, CodeInigter... và các hệ quản trị nội dung ...

[Cần giúp về mô hình **MVC trong PHP**](http://sinhvienit.net/forum/can-giup-ve-mo-hinh-mvc-trong-php.349544.html)

8 Tháng Mười Một 2014

[Xin giúp đỡ về OOP và **MVC** [**php**]](http://sinhvienit.net/forum/xin-giup-do-ve-oop-va-mvc-php.282108.html)

25 Tháng Chín 2013

[Xin trợ giúp về mô hình **MVC trong php**](http://sinhvienit.net/forum/xin-tro-giup-ve-mo-hinh-mvc-trong-php.245339.html)

17 Tháng Ba 2013

[Video Hướng dẫn xây dựng mô hình **MVC trong PHP**](http://sinhvienit.net/forum/video-huong-dan-xay-dung-mo-hinh-mvc-trong-php.22225.html)

30 Tháng Ba 2010

[Các kết quả khác từ sinhvienit.net](https://www.google.com.vn/search?q=mvc+trong+php+site:sinhvienit.net&espv=2&sa=X&ved=0ahUKEwiFx6y04OTSAhVJEbwKHajJDjcQrQIIWCgEMAg)

### [Ứng dụng mô hình MVC trong lập trình PHP - daotaolaptrinh](http://daotaolaptrinh.edu.vn/ung-dung-mo-hinh-mvc-trong-lap-trinh-php)

daotaolaptrinh.edu.vn/ung-dung-mo-hinh-mvc-trong-lap-trinh-php

Mô hình **MVC** được sử dụng nhiều **trong** lập trình nhữn ứng dụng với ASP.net. Trên thực tế mô hình**MVC** có thể được ứng dụng **trong** rất nhiều ngôn ngữ lập ...

### [Bài 12 : Mô Hình MVC | Học thiết kế web theo chuẩn. - Izwebz](https://www.izwebz.com/video-tutorials/bai-12-mo-hinh-mvc/)

https://www.izwebz.com/video-tutorials/bai-12-mo-hinh-mvc/

Như vậy có đôi lúc bạn đau cả đầu với những CV yêu cầu bạn biết **MVC** hoặc .... Cái ấy là do cái **php**designer auto., kakaka vì thế nên gõ từng chữ **trong** lập ...

### [Thể Hiện Mô Hình MVC Trên PHP :: Bài viết :: STDIO](https://www.stdio.vn/articles/read/159/the-hien-mo-hinh-mvc-tren-php)

https://www.stdio.vn/articles/read/159/the-hien-mo-hinh-mvc-tren-php

23 thg 1, 2015 - Bài trước tôi đã giới thiệu tới các bạn bài Design Pattern - **MVC** Pattern Hay Mô Hình**MVC**. Bài viết này tôi sẽ thể hiện mô hình **MVC** trên ngôn ...

### [Tìm hiểu mô hình MVC - Kiến thức Web](http://kienthucweb.net/tim-hieu-mo-hinh-mvc.html)

kienthucweb.net/tim-hieu-mo-hinh-mvc.html

5 thg 10, 2014 - Chúng ta có thể áp dụng mô hình **MVC** vào các dự án **trong** môi trường Windows, Linux… và sử dụng bất kỳ ngôn ngữ nào như **PHP**, ASP, ...

### [Bài 1 Lập trình website theo mô hình MVC - Xây dựng ứng dụng web](https://www.slideshare.net/tuoitrecomvn/slide-01-24160918)

https://www.slideshare.net/tuoitrecomvn/slide-01-24160918

11 thg 7, 2013 - Cách tổ chức mô hình **MVC** Ưu điểm của mô hình **MVC trong** việc phát ... 1 - Lập trình website theo mô hình **MVC** 15 Khái niệm **PHP** framework ...

### [Mô hình MVC trong PHP - những điều căn bản](http://lifeandline.com/mo-hinh-mvc-trong-php-nhung-dieu-can-ban.html)

lifeandline.com/mo-hinh-mvc-trong-php-nhung-dieu-can-ban.html

8 thg 9, 2014 - Khi mới bắt đầu viết những dòng code **PHP**, chúng ta chưa thể định hướng được một cấu trúc source như thế nào, lúc mới bước vào lập trình ...

### [Mô hình MVC trong PHP - phân tách chức năng](http://lifeandline.com/mo-hinh-mvc-trong-php-phan-tach-chuc-nang.html)

lifeandline.com/mo-hinh-mvc-trong-php-phan-tach-chuc-nang.html

9 thg 9, 2014 - Bài trước, chúng ta đã có một bước khởi động nhỏ **trong** việc giải quyết các vấn đề để đi đến mô hình **MVC**, **trong** bài viết này chúng ta sẽ giải ...

### [Giới thiệu về mô hình MVC trong PHP – Itvnsoft web developer](http://itvnsoft.com/gioi-thieu-ve-mo-hinh-mvc-trong-php/)

itvnsoft.com/gioi-thieu-ve-mo-hinh-mvc-trong-php/

**Trong** lập trình web hiện đại thì không thể không nhắc tói mô hình **MVC**, đây là mô hình phổ biến nhất hiện nay, hầu hết tất các framework nổi tiếng như Laravel, ...

### [[Codeigniter Framework] Phần 1: Mô hình MVC trong Codeigniter PHP ...](https://viblo.asia/chukimthang94/posts/MJykjQjqePB)

https://viblo.asia/chukimthang94/posts/MJykjQjqePB

21 thg 9, 2016 - Giới thiệu Codeigniter là 1 framework viết bằng ngôn ngữ lập trình **PHP** dựa trên mô hình **MVC**. Nó cho phép các nhà phát triển xây dựng các ...

### [[BÀI 28] MÔ HÌNH MVC TRONG PHP - Tự Học Lập Trình PHP](http://laptrinhphp.com.vn/bai-28-mo-hinh-mvc-trong-php/)

laptrinhphp.com.vn › PHP

[BÀI 28] MÔ HÌNH **MVC TRONG PHP**. Để cho việc viết code các trang web dễ nhìn và đơn giản khi viết cũng như dễ sửa chữa khi nâng cấp. Khi viết code, các ...

### [MVC trong PHP | Trang 1, ToiiT.com | Chia sẻ kiến thức lập trình php ...](http://toiit.com/danh-muc/mvc-trong-php-24)

toiit.com/danh-muc/mvc-trong-php-24

12 thg 3, 2017 - Trong lập trình web mô hình MVC là 1 mô hình chuẩn chung, được sử dụng rộng rãi, mô hình **MVC trong php** hay còn thường được gọi là các ...

content 1

# **Bài 01: MVC PHP - Mô hình MVC là gì?**

Đăng bởi: TheHalfHeart- Vào ngày: 31-03-2015- View: 47096

* [BÀI HỌC](http://freetuts.net/mvc-php-mo-hinh-mvc-la-gi-354.html#)
* [NEXT](http://freetuts.net/mvc-php-xay-dung-cau-truc-folder-355.html)

Trong bài này mình sẽ giới thiệu với các bạn mô hình MVC là gì? và đây cũng là bài mở đầu cho project MVC căn bản dành cho người mới bắt đầu tìm hiểu về MVC. Trong serie này mình sẽ làm một demo nho nhỏ về MVC xây dựng chức năng quản lý tin tức.

Hiện nay trên mạng có khá nhiều serie về MVC và họ viết rất nhiều cách khác nhau, riêng đối với serie này của mình thì không cần dùng dao to búa lớn làm gì mà chỉ hướng dẫn sử dụng tạo project theo mô hình MVC và thông qua project này các bạn sẽ hiểu được ý nghĩa của từng tầng và như vậy là mình đã thành công rồi. Còn bây giờ ta vào bài học chính nhé.

Nội dung chính

* [1. Mô hình MVC là gì?](http://freetuts.net/mvc-php-mo-hinh-mvc-la-gi-354.html#goto-h2-0)
* [Bài liên quan](http://freetuts.net/mvc-php-mo-hinh-mvc-la-gi-354.html#goto-h2-1)
* [2. Ưu điểm và nhược điểm mô hình MVC](http://freetuts.net/mvc-php-mo-hinh-mvc-la-gi-354.html#goto-h2-2)
* [3. Luồng xử lý trong mô hình MVC](http://freetuts.net/mvc-php-mo-hinh-mvc-la-gi-354.html#goto-h2-3)
* [4. lời kết](http://freetuts.net/mvc-php-mo-hinh-mvc-la-gi-354.html#goto-h2-4)

## **1. Mô hình MVC là gì?**

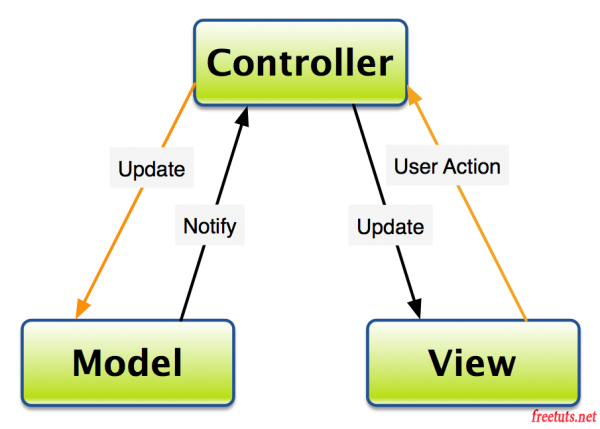
Từ hồi đi học mình đã tiếp xúc với mô hình 3 lớp, đến khi ra trường thì mình mới tiếp xúc tới mô hình MVC và mình bắt đầu tìm hiểu từ đó. Kể ra lúc đó cũng ngu, mà ngu một phần thôi mà vì nghèo nhiều hơn, mình học công nghệ thông tin nhưng không có điều kiện vật chất để sở hữu máy tính sớm nên không có cơ hội học online được, mãi khi ra trường mới được mở rộng tâm mắt.

MVC là chữ viết tắt của Model - View - Controller, đây là một mô hình kiến phần mềm được tạo ra với mục đích quản lý và xây dựng dự án phần mềm có hệ thống hơn. Mô hình này được dùng khá rộng rãi và đặc biệt là trong các ngôn ngữ lập trình web. Trong PHP hiện tại có khá nhiều Framework và tất cả đều xây dựng từ mô hình MVC, từ đó bạn có thể thấy sự quan trọng của nó như thế nào rồi đấy.

Trong mô hình này thì:

* Model: có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu và controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View
* View: có nhiệm vụ tiếp nhận dữ liệu từ controller và hiển thị nội dung sang các đoạn mã HTML, bạn có thể hiểu nôm na đây người ta còn gọi là thành phần giao diện.
* Controller: đóng vài trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client

Để rõ ràng hơn thì bạn xem hình dưới đây:



Nhìn vào mô hình này các bạn thấy giữa model và view không hề có mối liên hệ mà nó sẽ thông qua controller để giao tiếp với nhau. Hiện trên mạng có khá nhiều mô hình vẽ ra nhưng mình thấy nó quá rắc rối nên mình chọn hình này cho bạn dễ hiểu nhất.

## **2. Ưu điểm và nhược điểm mô hình MVC**

Bây giờ mình liệt kê một số ưu điểm và nhược điểm của mô hình MVC nhé.

Ưu điểm:

* Hệ thống phân ra từng phần nên dễ dáng phát triển
* Chia thành nhiều modun nhỏ nên nhiều người có thể làm chung dự án
* Vấn đề bảo trì cũng tương đối ok, dễ nâng cấp
* Dễ dàng debug trong quá trình xây dựng

Nhược điểm:

* Hệ thống sẽ chạy chậm hơn PHP thuần, tuy nhiên nó ko phải là vấn đề :D
* Xây dựng cầu kì và mất thời gian để xây dựng thư viện, cấu trúc

Xét về ưu và nhược thì rõ ràng nên xử dụng MVC phải không nào các bạn :D

## **3. Luồng xử lý trong mô hình MVC**

Ok bây giờ mình sẽ đưa ra một ví dụ về luồng xử lý trong mô hình MVC nhé.

Giả sử bạn đang xem một bài tuts trên website freetuts.net có URL là http://freetuts.net/rewrite-url-trong-codeigniter-344.html thì hệ thống MVC sẽ xử lý như sau:

* Bước 1: Dựa vào yêu cầu của bạn là xem bài viết có id=344 nên controller sẽ gọi tới một hàm lấy dữ liệu theo id trong model
* Bước 2: Sau khi có dữ liệu controller sẽ gửi qua View, lúc này view có nhiệm vụ xử lý dữ liệu và convert thành nhữn đoạn mã HTML
* Bước 3: Sau khi view kết thúc thì controller sẽ gửi trả nội dung HTML của view về cho client nên bạn sẽ xem được nội dung của bài tus có id=344

Cũng hơi rắc rối nhưng hy vọng bạn hiểu :3

## **4. lời kết**

Đây là bài đầu tiên trong serie xây dựng project MVC nên mình chỉ giới thiệu khái niệm mô hình MVC là gì, các thành phân cơ bản trong mô hình MVC như model, controller và view. Bài này về khái niệm mình viết ở dạng hiểu riêng nên có thể nó không giống với những bài ở các trang khác nhưng chung quy lại nó cũng giống nhau cả thôi. Bài tiếp ta học cách [xây dựng cấu trúc folder trong mô hình MVC](http://freetuts.net/mvc-php-xay-dung-cau-truc-folder-355.html), chúc bạn học tốt

content 2

## **Tìm hiểu về mô hình MVC (Model, Controller, View) trong PHP**

* / 15 Comments.[03 TH2](http://kungfuphp.com/php-nang-cao/tim-hieu-ve-mo-hinh-mvc-trong-php.html)/
* [Kungfu PHP](http://kungfuphp.com/author/programming-itergmail-com)/
* [PHP nâng cao](http://kungfuphp.com/category/php-nang-cao)

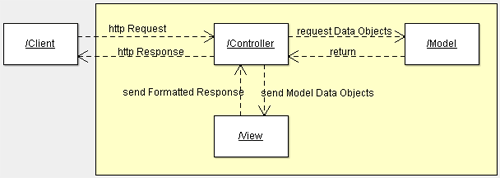
### **1. Giới thiệu về mô hình MVC trong PHP (Model, Controller, View)**

Mô hình MVC (model, controller, view) là mô hình chuẩn cho ứng dụng web được sử dụng nhiều nhất ngày nay. Mô hình MVC được sử dụng lần đầu tiên trong Smalltalk, sau đó được sử dụng phổ biến trong ngôn ngữ lập trình Java. Hiện nay, đã có hơn hàng chục PHP framework dựa trên mô hình này.

Bạn biết đấy, mô hình MVC hiện nay rất phổ biến trong các framework PHP, nhưng thực sự rất khó để tìm một bài viết với hướng dẫn chi tiết kèm theo những ví dụ đơn giản để chúng ta có thể hiểu được về nó. Đó là mục đích của hướng dẫn này

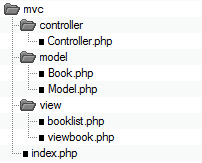
Mô hình MVC là viết tắt của 3 chữ Model, View, Controller. Mô hình này tách một ứng dụng web ra làm 3 thành phần đảm nhiệm chức năng tách biệt, thuận tiện cho việc xử lý và bảo trì

* Model : Chịu trách nhiệm quản lý dữ liệu, nó lưu trữ và truy xuất các thực thể từ cơ sở dữ liệu như mysql, sql server, postresSQL,… đồng thời chưa các logic được thực thi bởi ứng dụng
* View : Chịu trách nhiệm hiển thị dữ liệu đã được truy xuất từ model theo một format nào đó theo ý đồ của lập trình viên. Cách sử dụng của View tương tự như các module templates thường thấy trong các ứng dụng web phổ biến như WordPress, Joomla,…
* Controller : trung gian, làm nhiệm vụ xử lý cho model và view tương tác với nhau. Controller nhận request từ client, sau đó gọi các model để thực hiện các hoạt động được yêu cầu và gửi ra ngoài View. View sẽ chịu trách nhiệm format lại data từ controller gửi ra và trình bày dữ liệu theo 1 định dạng đầu ra (html).
* Hình minh họa dưới đây sẽ cho bạn thấy rõ cách mô hình MVC hoạt động



### **2. Từng bước thực hành ví dụ về mô hình MVC**

Đầu tiên, trong thư mục gốc của website ở local (với wamp thì là www, với xampp thì là htdocs), bạn tạo một folder đặt tên là mvc với cấu trúc thư mục như sau :



#### **a. Controller**

Đầu tiên chúng ta sẽ làm việc với controller, đây là nơi đầu tiền nhận các yêu cầu (requests), phân tích yêu cầu, khởi tạo và gọi model, sau đó nhận các hồi đáp (response) từ model và gửi ra các lớp giao diện (view). Trong thực tế Controller được gọi từ điểm đầu vào của ứng dụng là tập tin index.php. Tập tin này sẽ giao toàn bộ các yêu cầu gửi từ client cho controller xử lý.

Trong tập tin index.php ta thêm gõ đoạn code sau:

PHP

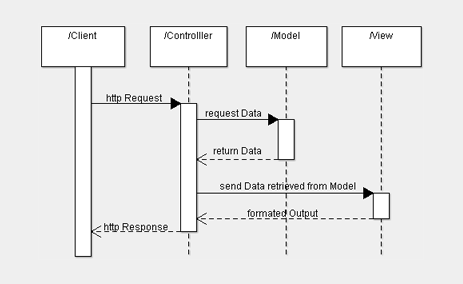
|  |  |
| --- | --- |
| 1  2  3  4  5 | // index.php file  include\_once("controller/Controller.php");    $controller = new Controller();  $controller->invoke(); |

Trong tập tin Controller.php của chúng ta sẽ tạo ra 1 function là invoke và một hàm khởi tạo contructor. Hàm contructor chịu trách nhiệm gọi và khởi tạo lớp Model. Trong hàm invoke sẽ quyết định data nào được phép trả ra từ model. Sau đó nó gọi model để lấy dữ liệu cần thiết, sau đó gửi dữ liệu ra view. Đoạn code cực kỳ đơn giản bên dưới sẽ cho bạn thấy điều đó. Lưu ý rằng, Controller không biết gì về cấu trúc database như thế nào cũng như là data gửi ra ngoài view sẽ có hình thù như thế nào bạn nhé. Nó chỉ có nhiệm vụ gọi model bởi request từ client và gửi dữ liệu ra ngoài view thôi nhé

PHP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | include\_once("model/Model.php");    class Controller {  public $model;    public function \_\_construct()  {  $this->model = new Model();  }    public function invoke()  {  if (!isset($\_GET['book']))  {  // nếu không có quyển sách nào được yêu cầu, thì hiển thị toàn bộ sách  $books = $this->model->getBookList();  include 'view/booklist.php';  }  else  {  // hiên thị sách được yêu cầu  $book = $this->model->getBook($\_GET['book']);  include 'view/viewbook.php';  }  }  } |

Mô hình dưới đây minh họa dễ hiểu hơn cho bạn



#### **b. Model**

Model đại diện cho dữ liệu và logic của ứng dụng, thường hay gọi là business logic. Model có trách nhiệm :

– Thêm, xóa sửa dữ liệu,…

– Là nơi thực thi logic nghiệp vụ của ứng dụng. Đối với những bạn mới tìm hiểu về MVC, chúng ta thường *nhầm lẫn và thực thì logic của nghiệp vụ ở bên trong Controller hoăc View*

Ví dụ dưới đây cho chúng ta thấy lớp model được đại diện bởi 2 class là : “Model” và “Book”.

PHP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | include\_once("model/Book.php");    class Model {  public function getBookList()  {  // Dưới đây mình gắn cứng 1 array để mô phỏng dữ liệu trong database  return array(  "Jungle Book" => new Book("Jungle Book", "R. Kipling", "A classic book."),  "Moonwalker" => new Book("Moonwalker", "J. Walker", ""),  "PHP for Dummies" => new Book("PHP for Dummies", "Some Smart Guy", "")  );  }    public function getBook($title)  {  //Gọi function getBooklist để lấy dữ liệu cần thiết  //Trong thực tế ở đây bạn sẽ dùng câu truy vấn select vào trong database  $allBooks = $this->getBookList();  return $allBooks[$title];  }      } |

Class Book trong file Book.php

PHP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | class Book {  public $title;  public $author;  public $description;    public function \_\_construct($title, $author, $description) {  $this->title = $title;  $this->author = $author;  $this->description = $description;  }  } |

#### **c. View**

View chịu trách nhiệm định đạng lại dữ liệu được truyền ra từ model. Dữ liệu được truyền ra có thể có nhiều định dạng khác nhau tùy vào cách người lập trình xử lý như là xml, json, array,..

Lưu ý là bạn đừng nên nhập nhằng, khó hiểu giữa view và template. Khi nhận được dữ liệu với định dạng nhất định từ hệ thống, tiếp tục view sẽ làm thao tác chuyển đổi dữ liệu thành một cấu trúc html cho người dùng thấy được. Thông thường 1 controller sẽ chỉ định gửi dữ liệu đến 1 view cụ thể. Chẳng hạn với công việc hiển thị tài khoản người dùng, thì Controller “display account” sẽ gọi đến lớp view “display account”. Tại đây, lớp view sẽ sử dụng 1 template có sẵn trong hệ thống để render ra các trang html. Mẫu template có sẵn này xử dụng lại những phần cố định của website như header, footer, menu,..

Đoạn code dưới đây cho ta 2 view dùng để hiển thị 1 quyển sách và nhiều quyền sách

viewbook.php

PHP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <html>  <head></head>    <body>    <?php    echo 'Title:' . $book->title . '<br/>';  echo 'Author:' . $book->author . '<br/>';  echo 'Description:' . $book->description . '<br/>';    ?>    </body>  </html> |

booklist.php

PHP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <html>  <head></head>    <body>    <table>  <tbody><tr><td>Title</td><td>Author</td><td>Description</td></tr></tbody>  <?php    foreach ($books as $title => $book)  {  echo '<tr><td><a href="index.php?book='.$book->title.'">'.$book->title.'</a></td><td>'.$book->author.'</td><td>'.$book->description.'</td></tr>';  }    ?>  </table>    </body>  </html> |

Các ví dụ trên của mình chính là cái nền cơ bản, cách hoạt động giống với các MVC Framework phổ biến hiện nay như CI, Zend,…Hy vọng qua bài viết này, các bạn có cái nhìn rõ ràng hơn về mô hình MVC và học các framework nhanh hơn. Kết thúc bài viết mình sẽ liệt kê một số ưu điểm của MVC :

* Model và View tách biệt, nên ứng dụng linh hoạt hơn, dễ bảo trì hơn các website truyền thống, một nùi code trong 1 file
* Model và View có thể coi là tách biệt, độc lập. Có thể 1 máy chủ A chỉ làm nhiệm vụ xử lý Model, back-end, một máy chủ B chỉ làm nhiệm vụ xử lý giao diện, front-end,.

content 3

# **Mô Hình MVC**

VMC là mô hình quen thuộc với các bạn đã học framework hay đã tìm hiểu.Đó là điều không thể thiếu để chuẩn bị bắt đầu vào học 1 framework nào đó.Hôm nay mình sẽ hướng dẫn cho các bạn 1 mô hình MVC tự động load controller, model

– Ưu điểm :là tự động load [controller](http://hocphp.info/controller-va-urls-trong-codeigniter-framework/) và [model](http://hocphp.info/model-trong-codeigniter/) gửi dữ liệu và hiển thị trong [view](http://hocphp.info/view-trong-codeigniter/) gần giống như trong các framework

– Các yêu cầu để thực hành được bài này các bạn lên xem qua các bài viết ở dưới để dễ tiếp cận hơn:

1. [Lập trình hướng đối tượng (OOP)](http://hocphp.info/lap-trinh-huong-doi-tuong-oop/)
2. [Phương thức MAGIC và tính chất kế thừa trong PHP](http://hocphp.info/phuong-thuc-magic-va-tinh-chat-ke-thua-trong-php/)
3. [Phương thức và thuộc tính tĩnh](http://hocphp.info/phuong-thuc-va-thuoc-tinh-tinh/)

## **Cấu trúc mô hình MVC**

– MVC chia ứng dụng của bạn làm 3 phần:

1. Model: chịu trách nhiệm quản lý dữ liệu, mọi tác vụ tương tác với cơ sở dữ liệu(CSDL) của bạn đều nằm ở đây.
2. View: chịu trách nhiệm hiển thị dữ liệu .
3. Controller: điều khiển view và model làm việc với nhau. Controller nhận request từ client, gọi model thực thi tương ứng, và gởi trả dữ liệu lại cho client bằng view, view định dạng dữ liệu để hiển thị cho người dùng.

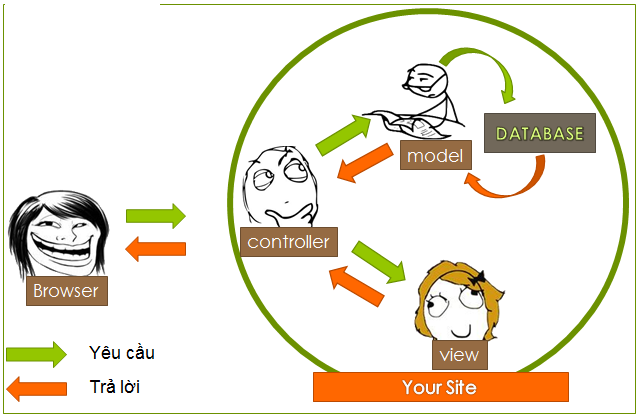
## **Mô hình xử lý dữ liệu của MVC framework**

– Người dùng (client) dùng web browser, gởi request đến Server , mọi request gởi đến Server đều được điều hướng đến file index.php.

– Tại index.php chúng ta thực thi Controller, Controller có nhiệm vụ bắt request, phân tích request để gọi Model xử lý và đổ dữ liệu ra View tương ứng.

– Sau đó hiển thị kết quả View ra trình duyệt cho người dùng.

Các bạn có thể xem hình ảnh minh họa phía dưới để có cái nhìn khái quát về cách hoạt động của mô hình này



Mô hình MVC

– ví dụ về chức năng đăng nhập bằng mô hình MVC

— Các bạn tạo class Bootstrap trong file Bootstrap.php với nội dung sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | <?php  class Bootstrap{  public function \_\_construct(){  if(isset($\_GET['url'])){  //lấy chuỗi url  $url=rtrim($\_GET['url'],"/");  //chuyển chuỗi url và dạng mảng  $url=explode('/',$url);  $c=$url[0];//lấy tên controller  }else{  $c="index";  }  //load file controller  $file\_controller=\_\_CONTROLLER\_PATH.$c."\_Controller.php";  if(file\_exists($file\_controller)){  require\_once($file\_controller);  }else{  require\_once(\_\_CONTROLLER\_PATH."error\_Controller.php");  $controller=new error();  $controller->index();  return false;  }  //khởi tạo class controller tương ứng  $name\_controller=$c."\_Controller";  $controller=new $name\_controller;  //tự động load model của controller tương ứng  $controller->LoadModel($c);  if(isset($url[2])){  //nếu tồn tại tham số của phương thức(action) trong controller sẽ chạy phương thức với tham số này  $controller->{$url[1]}($url[2]);  }else{  //nếu tồn tại phương thức(action) trong controller sẽ chạy phương thức này  if(isset($url[1])){  $controller->{$url[1]}();  }else{ //mặc định sẽ chạy phương thức index  $controller->index();  }  }  }  }  ?> |

+ file này có tác dụng tự động load controller và model khi người dùng tương tác tới trang

+ Lưu ý: Đặt tên file controller và model có phần đầu giống nhau,và tên file giống tên class,ví dụ user\_Controller.php thì class tương tứng sẽ là class user\_Controller và model sẽ là user\_Model.php class user\_Model.Như vậy trong view sẽ có thư mục tương ứng là thư mục user,

— Trang index sẽ khởi tạo Bootstrap như sau :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <?php  //define path  require\_once("config/define\_path.php");  require\_once("config/define\_database.php");  //end define path  require\_once(\_\_LIB\_PATH."Session.php");  require\_once(\_\_LIB\_PATH."Controller.php");  require\_once(\_\_LIB\_PATH."View.php");  require\_once(\_\_LIB\_PATH."Model.php");  require\_once(\_\_LIB\_PATH."Bootstrap.php");  $bootstrap= new Bootstrap;  ?> |

—Thư mục dự án của mình có tên là mvc,lên khhi người dùng truy cập vào trang http://localhost/vmc/user/loginclass Bootstrap sẽ thực hiện action login trong controller user\_Controller.php và load model user\_Model.php để sử dụng

— Sau đây là ví dụ login với mô hình này :

+ Chuẩn bị :tạo bảng user (user\_name, pass)

+ file controllers/user\_Controller.php :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | <?php  class user\_Controller extends Controller{  public function \_\_construct(){  parent::\_\_construct();  }  public function index(){  $this->view->render("user/index");  }  public function login(){  if(isset($\_POST['login'])){  if($\_POST['user']=="" or $\_POST['password']==""){  $this->view->msg="Hay dien user hay pass";//gửi thông bào tới trang view  $this->view->render("user/login");  }else{  if($this->model->login()==true){  Session::set("login",true);  Session::set("user",$\_POST['user']);  $this->view->redirect();  }else{  $this->view->redirect("user/login");  }  }  }else{  $this->view->render("user/login");  }    }  public function logout(){  Session::set("login",false);  Session::unset\_session("user");  $this->view->redirect();  }  }  ?> |

—File models/user\_Model.php :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <?php  class user\_Model extends Model{  public $table="user";  public function \_\_construct(){  parent::\_\_construct();  }  public function login(){  $user=$\_POST['user'];  $pass=md5($\_POST['password']);  $where=array('user\_name'=>$user,'pass'=>$pass);  $this->select($this->table,$where);  if($this->num\_rows()==0){  return false;  }else{  return true;  }  }    }  ?> |

—file Views/user/login.php : sẽ hiển thị lỗi lấy từ controller nếu có và form đăng nhập

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <h2>Trang login.....</h2>  <hr />  <div class="error" style="background:#FC6;color:red">  <?php  if(isset($this->msg)){  echo $this->msg;  }  ?>  </div>  <form action="" method="post">  <label>User </label><input name="user" type="text" /><br />  <label>Password </label><input name="password" type="text" /><br />  <label></label><input type="submit" value="login" name="login"/><br />  </form> |

— Các bạn có thể down về để hiểu rõ thêm : <http://www.mediafire.com/?3d59b8eb70x27dx>

Kết thúc bài viết: Qua bài này các bạn đã có cái nhìn tổng quát và cách hoạt động trong mô hình MVC rồi phải không.Đây là 1 phần quan trọng để các bạn bắt đầu vào học 1 framework nào đó.