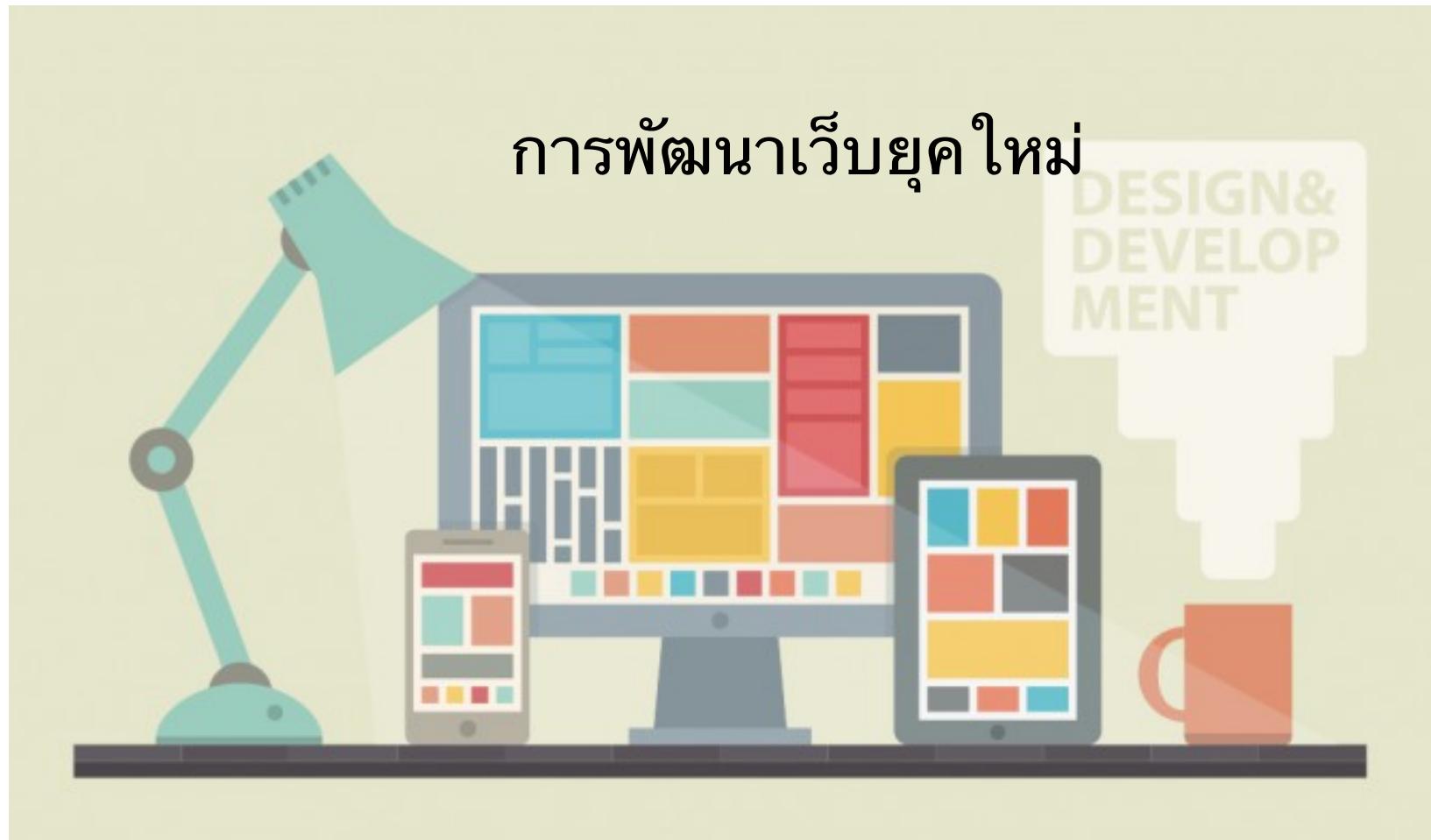


# Modern Web Development



By IWA Labs Thailand Co.,Ltd.  
[www.iwalabs.com](http://www.iwalabs.com)

# เกี่ยวกับวิทยากร

ชื่อ: วิเชียรนิตช์เวช  
ชื่อเล่น: วากกี้

Email:[waycs16@gmail.com](mailto:waycs16@gmail.com)

Facebook:[www.facebook.com/waywatee](https://www.facebook.com/waywatee)

Line:[hypedway](#)

Website:[www.expstudio.net](http://www.expstudio.net)

“Long Experience Senior Web Developer at IWA Labs Thailand  
and Owner of Laravel Blog”

# บทที่ 3 jQuery

- เป็น JavaScript library ที่จำเป็นต้องรู้วิธีการใช้ในปัจจุบัน
- การใช้งานหลักคือการเข้าถึง HTML element ได้โดยง่าย
- ดาวน์โหลดที่ <http://jquery.com/>
- ศึกษาเพิ่มเติมที่ <http://www.thaicreate.com/jquery.html>

# เนื้อหาอบรม

สามารถเข้าศึกษาเนื้อหาออนไลน์ได้ที่ [www.expstudio.net/laravel](http://www.expstudio.net/laravel)

|         | หัวข้อ               | ระยะเวลา  |
|---------|----------------------|-----------|
| ภาค 1   | การพัฒนาเว็บสมัยใหม่ | 2 ชั่วโมง |
| บทที่ 1 | ภาพรวม               | 15 นาที   |
| บทที่ 2 | Twitter's Bootstrap  | 30 นาที   |
| บทที่ 3 | jQuery               | 30 นาที   |
| บทที่ 4 | แหล่งหากองฟรี        | 15 นาที   |

# บทที่ 6 เริ่มทำเว็บแรก

- ตั้งค่าการเชื่อมต่อฐานข้อมูล MySQL
  - กำหนดค่าให้กับ อาร์เรย์ connections mysql

```
1 return array(
2     .....
3     'connections' => array(
4         .....
5         'mysql' => array(
6             'driver'      => 'mysql',
7             'host'        => 'localhost', // Host Database
8             'database'    => 'db_tutorial_laravel', // Name Database
9             'username'    => 'root', // Username Database
10            'password'   => '1234', // Username Database
11            'charset'    => 'utf8', // Charset Database
12            'collation'  => 'utf8_unicode_ci',
13            'prefix'     => '',
14        ),
15        .....

```

# เนื้อหาอบรม

| ภาค 2    | การพัฒนาเว็บไซต์ PHP ด้วย Framework Laravel 4.2             | 8 ชั่วโมง   |
|----------|---|-------------|
| บทที่ 5  | องค์ประกอบของ Laravel                                       | 30 นาที     |
| บทที่ 6  | ติดตั้ง Laravel   | 30 นาที     |
| บทที่ 7  | เริ่มทำเว็บแรก  | 30 นาที     |
| บทที่ 8  | Url Routing คืออะไร ใช้งานอย่างไร                           | 30 นาที     |
| บทที่ 9  | สร้างส่วนแสดงผลบนหน้าเว็บที่เรียกว่า View                   | 30 นาที     |
| บทที่ 10 | สร้างฐานข้อมูลด้วย Migration, Schema Builder, Seeding       | 30 นาที     |
| บทที่ 11 | เชื่อมต่อฐานข้อมูลกับเว็บด้วย Model และ Eloquent ORM        | 30 นาที     |
| บทที่ 12 | การเชื่อมต่อข้อมูลเข้ากับหน้าแสดงผลด้วย Controller          | 30 นาที     |
| บทที่ 13 | การสร้าง View ด้วย blade template                           | 30 นาที     |
| บทที่ 14 | สร้างฟอร์มจัดการข้อมูล และ CRUD : Create Read Update Delete | 1.5 ชั่วโมง |

# เนื้อหาอบรม

|           |  |             |
|-----------|--|-------------|
| บทที่ 15  | การตรวจสอบข้อมูล Data Validation         | 15 นาที     |
| บทที่ 16  | การแบ่งหน้าแสดงข้อมูล หรือ Pagination    | 15 นาที     |
| บทที่ 17  | การอัพโหลดไฟล์ด้วย LaraClip              | 15 นาที     |
| บทที่ 18  | การใช้งาน Session                        | 15 นาที     |
| บทที่ 19  | การส่งเมล์ และการทำ Authentication       | 30 นาที     |
| บทที่ 20  | การติดตั้งเว็บบนโฮสท์                    | 30 นาที     |
| ภาค 3     | ลองของจริง                               | 3 ชั่วโมง   |
| บทพิเศษ 1 | สร้างเว็บไซต์ Portfolio แสดงผลงาน        | 1.5 ชั่วโมง |
| บทพิเศษ 2 | ใช้เทมเพลต Laravel 4 Startup Boilerplate | 1.5 ชั่วโมง |

# บทที่ 1 ภาพรวม

- เครื่องไม้เครื่องมือ
  - TextEditor : SublimeText3, Brackets
  - ไปที่ <http://www.sublimetext.com/> เพื่อทำการติดตั้ง
- ตัวช่วยสร้าง ui สวยและรองรับอุปกรณ์หลากหลาย HTML, CSS, and JS framework for Responsive Design
  - Twitter's Bootstrap
  - Zurb's Foundation
  - Sematic UI
  - Gumby Framework 2
  - เพิ่มเติมที่ <http://www.designil.com/web-design-css-framework-list.html>
- JavaScript Framework เพื่อควบคุมการแสดงผลให้ล้ำสมัย
  - jQuery

# บทที่ 1 ภาพรวม (ต่อ)

- ใช้ Template เพื่อให้เว็บเสร็จไวขึ้น
  - Bootstrap Template
  - HTML Template
  - [www.wrapbootstrap.com](http://www.wrapbootstrap.com)
  - [www.bootswatch.com](http://www.bootswatch.com)
  - [www.templatemonster.com](http://www.templatemonster.com)
- สร้างเว็บด้วย Framework เพื่อให้เป็นมาตรฐาน และง่ายต่อการดูแลรักษา ใช้คล่องกับเครื่องไว้ด้วย
  - Laravel
  - Yii
  - CodeIgniter
  - CakePHP

# บทที่ 2 Twitter's Bootstrap

Bootstrap

Getting started

CSS

Components

JavaScript

Customize

Expo

Blog



Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

[Download Bootstrap](#)

Currently v3.2.0

# องค์ประกอบหลักของ bootstrap

- CSS
  - ใช้กำหนดหน้าตาของ html element ให้สวยงามด้วยการตั้งชื่อ class หรือกำหนด class ที่ต้องการ
- Component
  - ชุดของคำสั่ง HTML และ CSS ที่เขียนไว้แล้วพร้อมใช้งานเพิ่มความรวดเร็วในการพัฒนาเว็บไซต์
- JavaScript
  - ชุดคำสั่ง JavaScript ที่ใช้ในการควบคุมการทำงานของเว็บไซต์ให้ดูมีอิทธิพล และเป็นระเบียบ

# การใช้งาน Bootstrap

Getting start bootstrap



ค้นหาด้วย Google

ต้องการค้นแม่นยำเพิ่ม

- <http://getbootstrap.com/getting-started/>
  1. ดาวน์โหลดจาก <http://getbootstrap.com/getting-started/>
  2. เรียกใช้ไฟล์ css และ javascript ใน html ของหน้าเว็บที่ต้องการใช้งาน

```
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">  
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">  
<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
```

# ตัวอย่าง

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap 101 Template</title>

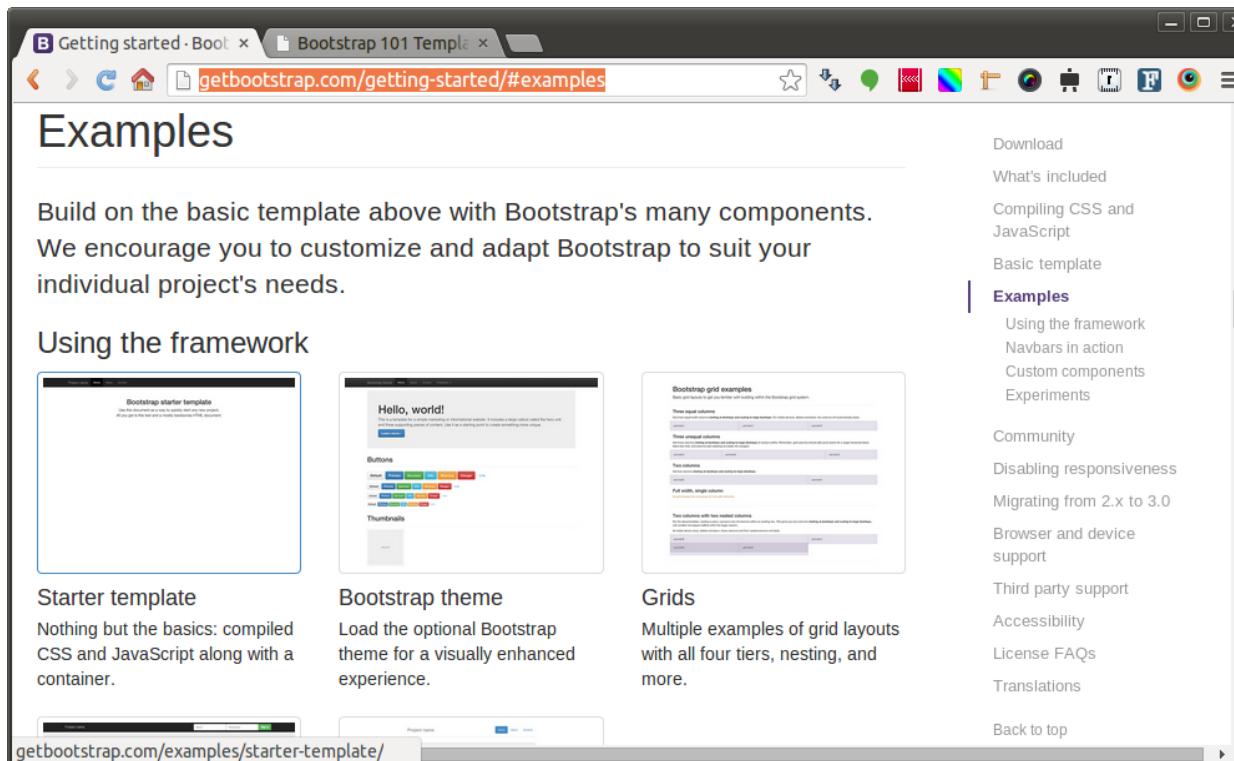
    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

  </head>
  <body>
    <h1>Hello, world!</h1>
    <p class="lead">Use this document as a way to quickly start any new project.<br> All you get is this text and a mostly barebones HTML document.</p>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

# ลองของจริง

- คลิ๊กไปที่examples
  - <http://getbootstrap.com/getting-started/#examples>
- คลิ๊กที่รูป ตัวอย่างแต่ละรูป แล้ว Ctrl+U เพื่อดู Code



FOLDERS

delovediamond

app

commands

config

controllers

admin

.gitkeep

BaseController.php

DiamondsController.php

FavoriteListsController.php

HomeController.php

MenusController.php

MergeController.php

NewsController.php

PagesController.php

ProductsController.php

PromotionsController.php

UploadController.php

UserController.php

database

lang

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="utf-8">
5          <meta http-equiv="X-UA-Compatible" content="IE=edge">
6          <meta name="viewport" content="width=device-width, initial-scale=1">
7          <meta name="description" content="">
8          <meta name="author" content="">
9          <link rel="icon" href="../../favicon.ico">
10
11      <title>Starter Template for Bootstrap</title>
12
13      <!-- Bootstrap core CSS -->
14      <link href="css/bootstrap.min.css" rel="stylesheet">
15
16      <!-- Custom styles for this template -->
17      <link href="starter-template.css" rel="stylesheet">
18
19      <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media
20          queries -->
21      <!--[if lt IE 9]>
22          <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></
23          script>
24          <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
25      <![endif]-->
26
27      </head>
28
29      <body>
```

<div class="navbar navbar-inverse navbar-fixed-top" role="navigation">

bootstrap\_example\_2.html

bootstrap\_example\_1.html



# บทที่ 3 jQuery

- เป็น JavaScript library ที่จำเป็นต้องรู้วิธีการใช้ในปัจจุบัน
- การใช้งานหลักคือการเข้าถึง HTML element ได้โดยง่าย
- ดาวน์โหลดที่ <http://jquery.com/>
- ศึกษาเพิ่มเติมที่ <http://www.thaicreate.com/jquery.html>

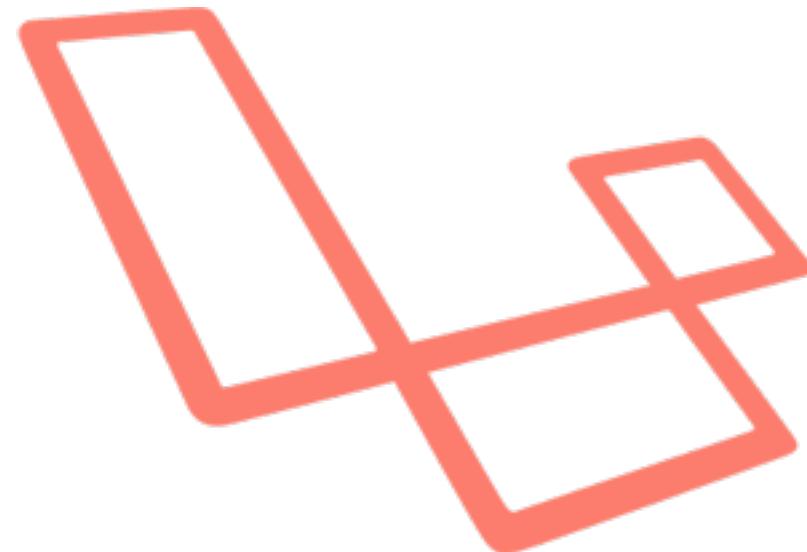
# บทที่ 4 แหล่งรวมของฟรี

- Wrapbootstrap.com
- Startbootstrap.com
- owlgraphic.com/owlcarousel
- Google.com
- GitHub.com
- SourceForge.com
- พยายามหาของฟรีที่มีคนทำไว้แล้วก่อนคิดจะทำเอง

# ภาค 2

การพัฒนาเว็บไซต์ HP  
ด้วย

Framework Laravel 4.2



# บทที่ 4 องค์ประกอบของ Laravel

- PHP5.4
- HTTP Server > Apache, built-in PHP server
- Database > Sqlite, MySql
- Composer
- Schema & Migration
- Model-View-Controller
- URL Routing
- RESTFul

# บทที่ 4 องค์ประกอบของ Laravel (ต่อ)

- Composer
  - คือเครื่องมือที่ช่วยในการติดตั้งส่วนของโค้ดที่มีคนสร้างเอาไว้ให้เราใช้แล้ว เช่น ระบบล็อกอิน การอัพโหลดไฟล์รูปภาพ
- Schema
  - หมายถึง โครงสร้างของข้อมูลในระบบที่เราจะพัฒนา เช่น User, Products, News
- Migration
  - หมายถึง การสร้าง Table ข้อมูลตาม Schema ที่กำหนดลงในฐานข้อมูลที่ต้องการ หรือการแก้ไข Table ที่มีอยู่แล้ว ในฐานข้อมูล

# บทที่ 4 องค์ประกอบของ Laravel (ต่อ)

- Model
  - ความหมายบางส่วนคือ เอาไว้แทน object หรือข้อมูลที่เราจะนำมาใช้งาน หรือจัดเก็บ หรืออาจจะมายถึง Table หลัก หรือ Entity เช่น User, Student, Course, Product, City
- View
  - คือส่วนที่แสดงผลข้อมูล อาจจะเป็นข้อมูลจาก Model หรือข้อมูลอะไรมีได้ หรืออาจจะทำหน้าที่รับ Input จาก Form ต่างๆเพื่อบันทึกลงไปยัง Model
- Controller
  - คือส่วนที่จะควบคุมการทำงานของ View หรือ เชื่อมระหว่าง View กับ Model เพื่อให้แสดงผลตามต้องการ หรือควบคุมการกระทำกับข้อมูลเช่น Insert Create Update Delete

# บทที่ 4 องค์ประกอบของ Laravel (ต่อ)

- URL Routing
  - คือการกำหนดว่าจะให้แสดงผล view หรือ controller หน้าไหน เช่น
    - myweb.com/blog แสดงรายการ log ทั้งหมด
    - myweb.com/blog/1 แสดงเฉพาะ log หมายเลข 1

# บทที่ 4 องค์ประกอบของ Laravel (ต่อ)

- RESTFul
  - เป็นสถาปัตยกรรมอย่างหนึ่ง ในการประยุกต์ใช้ protocal http ให้สื่อความหมาย เช่น
    - เรียก url : myweb.com/blog ผ่าน web browser ซึ่งประเภทของ request จะเป็น method GET อัตโนมัติ เราสามารถกำหนด URL Routing ไปที่ action หรือ function index ใน controller เพื่อแสดงผลข้อมูลจาก model Blog ทั้งหมดได้
    - บันทึก หรือ submit Form โดย action ของ form เป็น url: myweb.com/blog ประเภทของ request จะเป็น method POST ให้เอง โดยปกติจะมีความหมายเป็นการสร้าง object ใหม่

# บทที่ 4 องค์ประกอบของ Laravel (ต่อ)

- RESTFul (ต่อ)
  - Request Action มีดังนี้
    - GET : ใช้อ่านค่ามาแสดง อาจจะเป็นลิสต์ หรือรายการเดียว  
มักจะเชื่อมกับ controller > index หรือ show
    - POST : ใช้สร้าง object ใหม่ มักจะเชื่อมกับ controller >  
create
    - PUT : ใช้แก้ไข object เดิม มักจะเชื่อมกับ controller >  
update
    - DELETE : ใช้ลบหรือทำลาย object มักจะเชื่อมกับ  
controller > destroy

# บทที่ 5 ติดตั้ง Laravel

- ติดตั้ง PHP 5.5 ด้วย XAMMP for Windows
  - เข้าไปที่ <https://www.apachefriends.org> และทำการดาวน์โหลดไฟล์xampp-win32-1.8.3-4-VC11-installer.exe ตามรูปภาพด้านล่าง ติดตั้งตามขั้นตอน

## XAMPP Apache + MySQL + PHP + Perl

---

### What is XAMPP?

XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MySQL, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

**Download**  
Click here for other versions

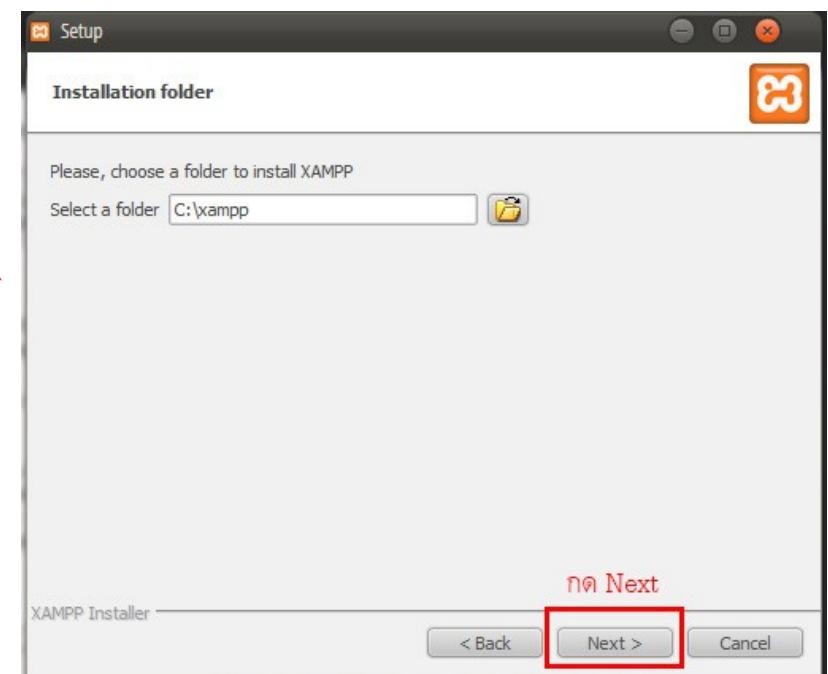
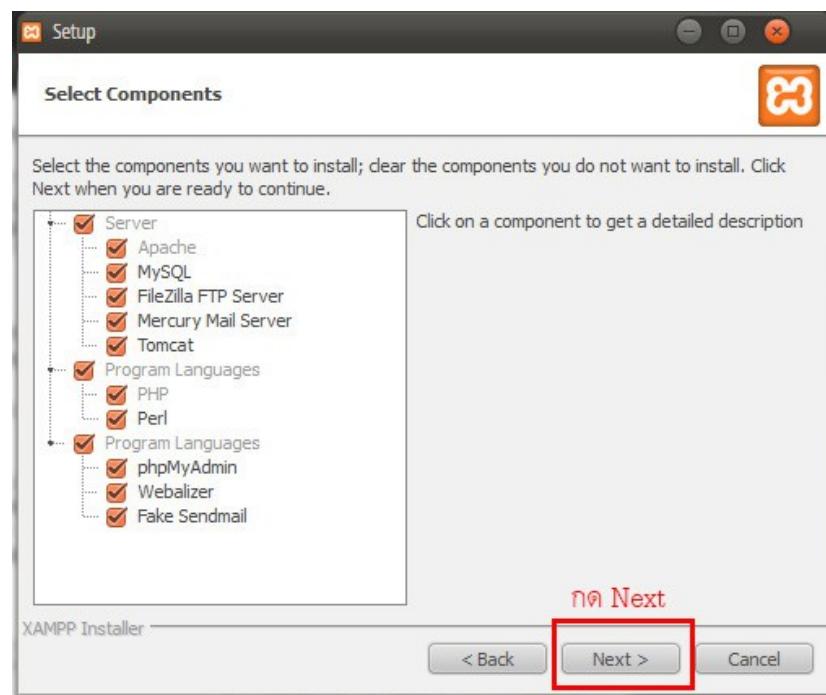
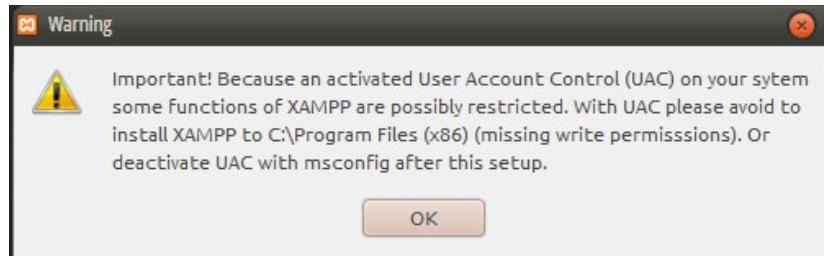
**XAMPP for Windows**  
v1.8.3 (PHP 5.5.11)

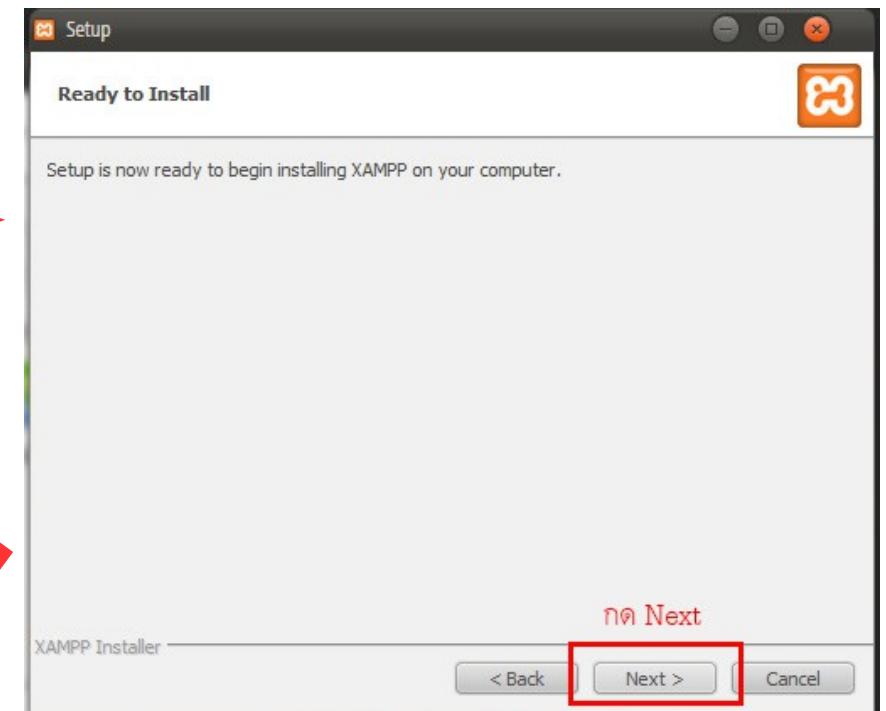
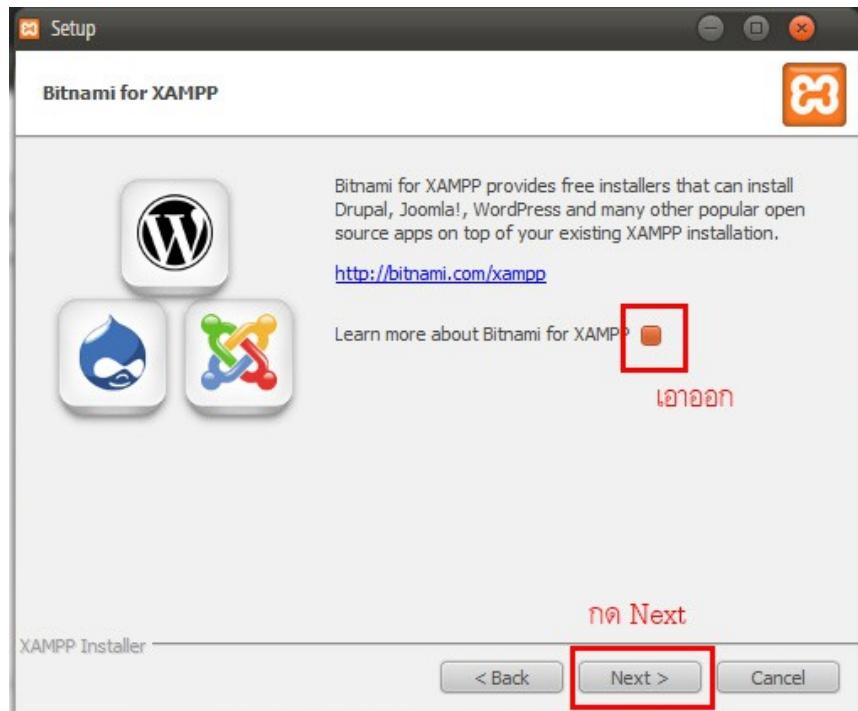
**XAMPP for Linux**  
v1.8.3 (PHP 5.5.11)

**XAMPP for Apple**  
v1.8.3 (PHP 5.5.11)

### Why use XAMPP?

- The most popular PHP dev package
- For Windows, Mac OS X & Linux
- Easy installation and configuration
- Completely free of charge





# บทที่ 5 ติดตั้ง Laravel

- การติดตั้ง PHP Composer
  - เข้าไปที่ <https://getcomposer.org/download/> และทำการดาวน์โหลดไฟล์ composer แบบ(1) และ(2) ตามรูปภาพด้านล่าง

## Windows Installer

The installer will download composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - It will install the latest composer version whenever it is executed.

## Manual Download

If you prefer to download the phar manually, here are the available versions:

### Latest Snapshot

- [1.0.0-alpha8](#) (2014-01-06)
- [1.0.0-alpha7](#) (2013-05-04)
- [1.0.0-alpha6](#) (2012-10-23)
- [1.0.0-alpha5](#) (2012-08-18)
- [1.0.0-alpha4](#) (2012-07-04)
- [1.0.0-alpha3](#) (2012-05-13)
- [1.0.0-alpha2](#) (2012-04-03)
- [1.0.0-alpha1](#) (2012-03-01)

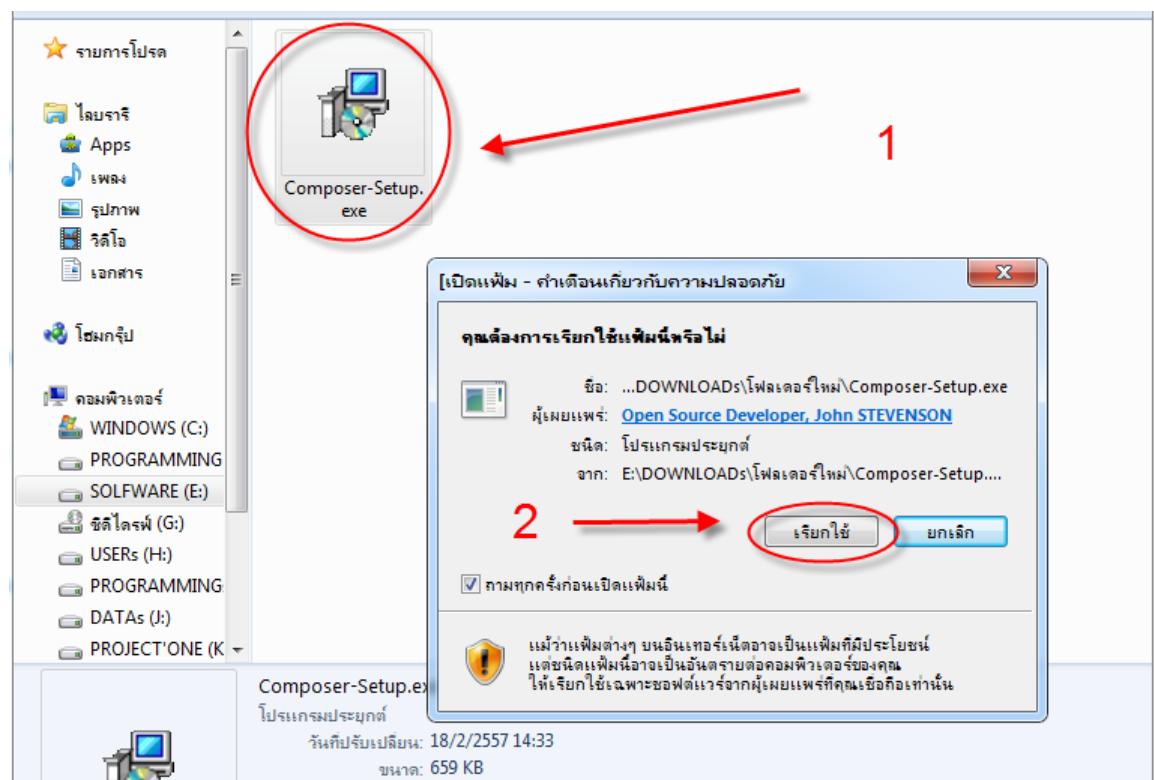
1

2

# บทที่ 5 ติดตั้ง Laravel

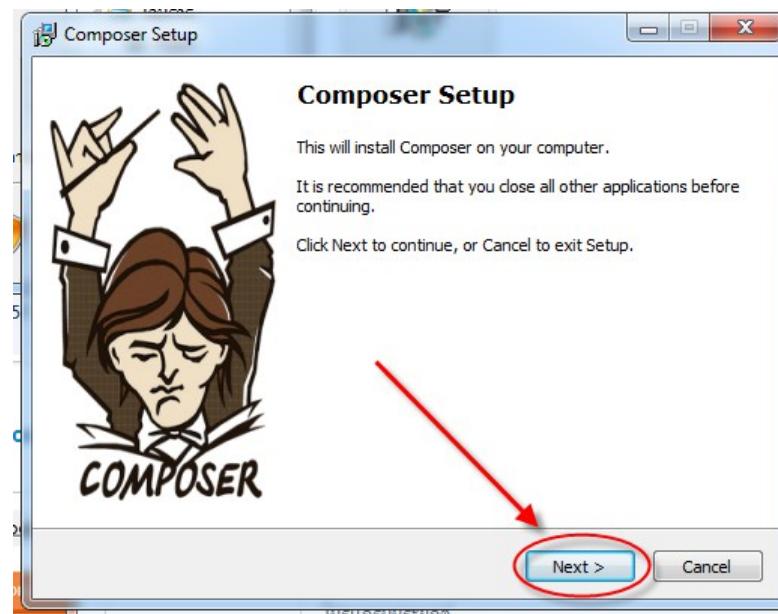
- การติดตั้ง PHP Composer

- ทำการติดตั้งโปรแกรม Composer โดยให้ดับเบิลคลิกไฟล์ "Composer (.exe)" ที่เราดาวน์โหลดมา (หมายเลข 1 รูปด้านบน)



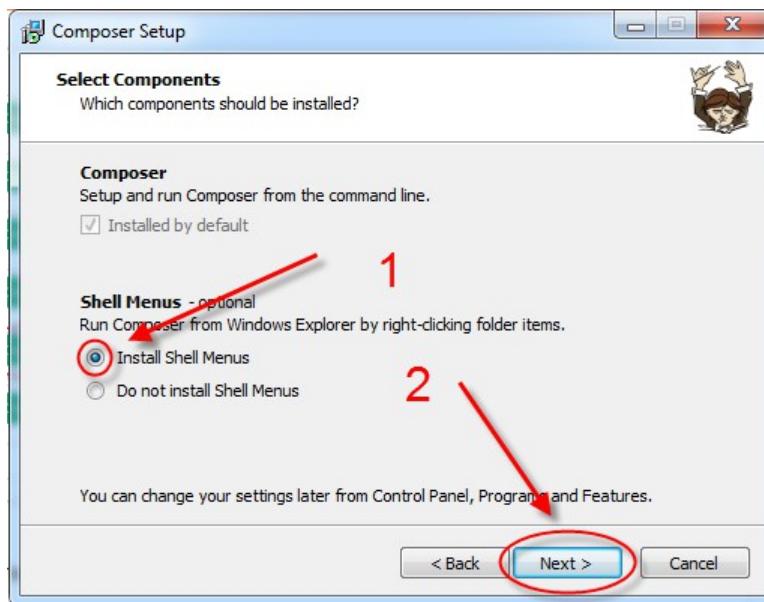
# บทที่ 5 ติดตั้ง Laravel

- การติดตั้ง PHP Composer
  - กดปุ่ม "Next" เพื่อติดตั้ง "Composer"



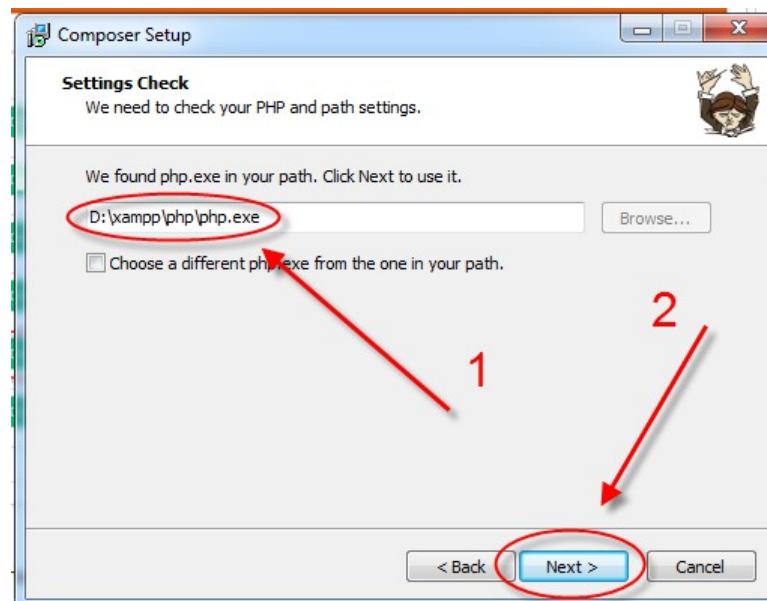
# บทที่ 5 ติดตั้ง Laravel

- การติดตั้ง PHP Composer
  - เลือกรายการ Install Shell Menu และคลิกปุ่ม "Next"



# บทที่ 5 ติดตั้ง Laravel

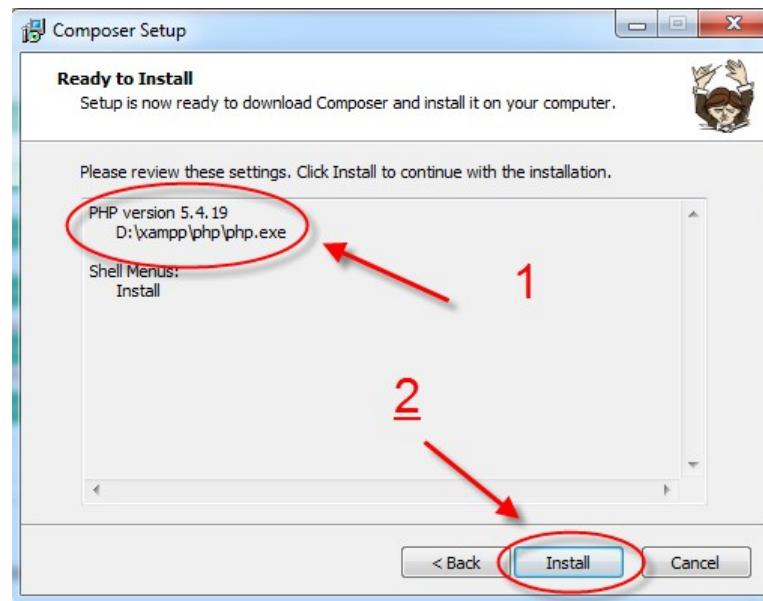
- การติดตั้ง PHP Composer
  - เลือกไดเรคทอรีที่เก็บไฟล์ "php.exe" ดาวน์โหลดติดตั้ง xampp servers และคลิกปุ่ม "Next"



# บทที่ 5 ติดตั้ง Laravel

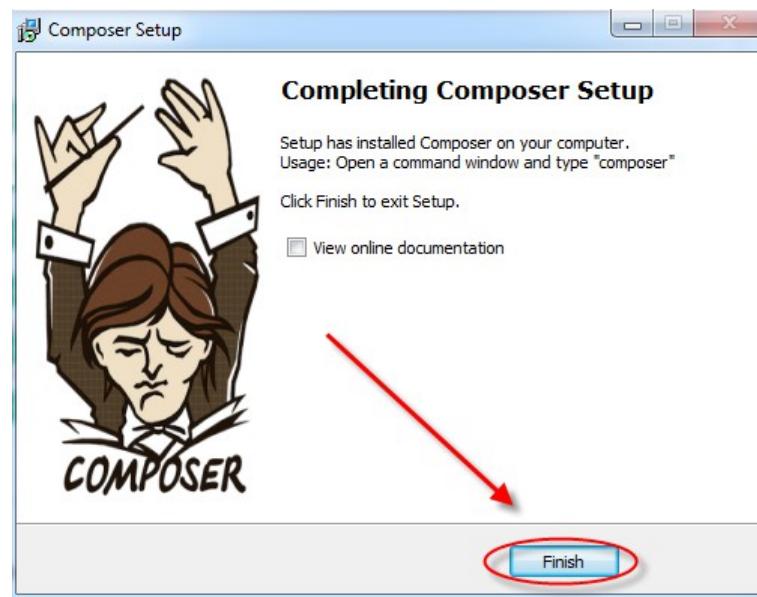
- การติดตั้ง PHP Composer

- แสดงรายละเอียดเวอร์ชันPHP"และตัวแทนไฟล์ไดเรอทอรีPHP"และคลิกปุ่ม"Install"เพื่อทำการติดตั้ง Composer



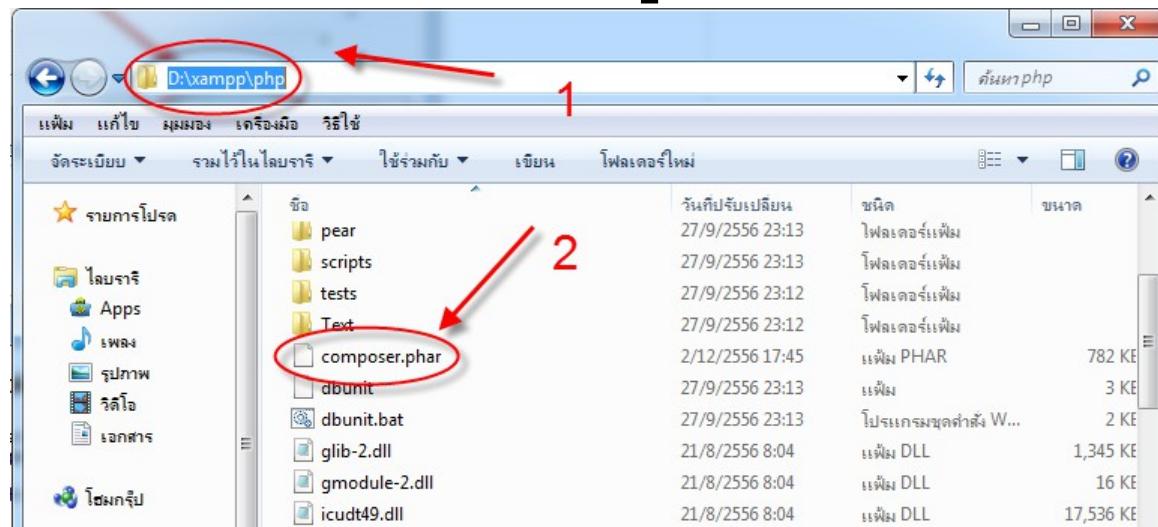
# บทที่ 5 ติดตั้ง Laravel

- การติดตั้ง PHP Composer
  - การติดตั้ง "Composer" สำเร็จแล้ว.. และกดปุ่ม "Finish"



# บทที่ 5 ติดตั้ง Laravel

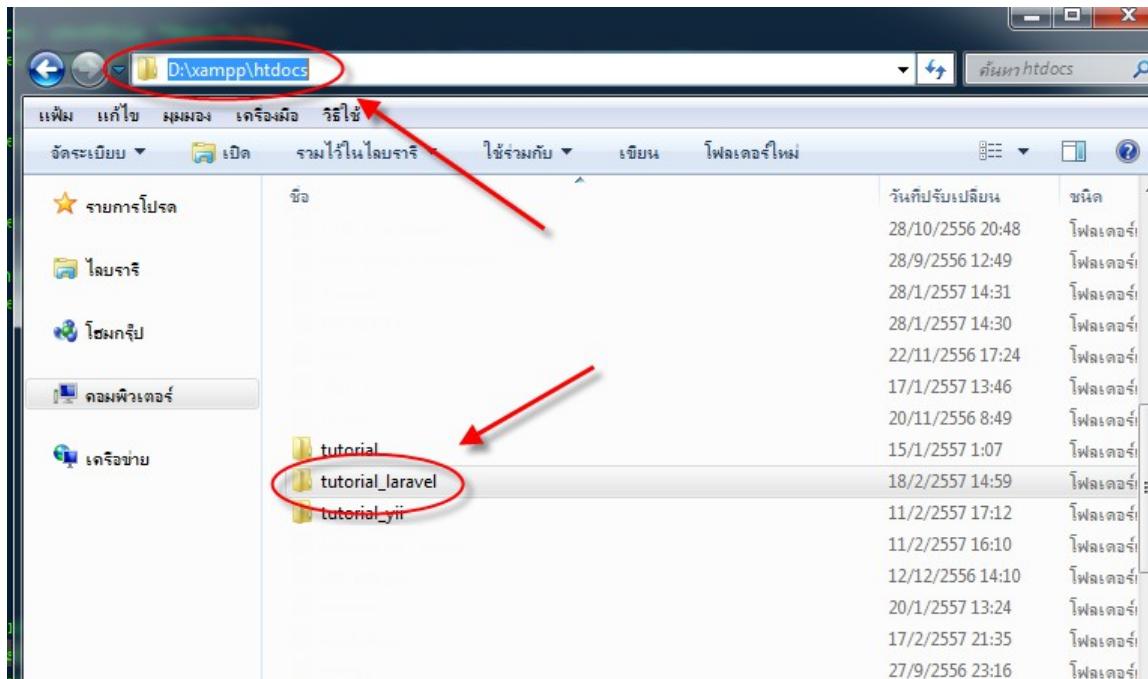
- การติดตั้ง PHP Composer ขั้นตอนสุดท้าย
  - ตรวจสอบไฟล์ Composer (.phar) ที่เราได้ติดตั้ง โดยให้เข้าไปที่ "ไดเรกทอรี(1)" ที่เราทำการติดตั้ง (ตามข้อ 5) และเมื่อเข้าไปยังไดเรกทอรี ดังกล่าวแล้ว เราจะพบไฟล์ชื่อว่า "composer.phar" ตามรูป(2) ด้านล่าง



# บทที่ 5 ติดตั้ง Laravel

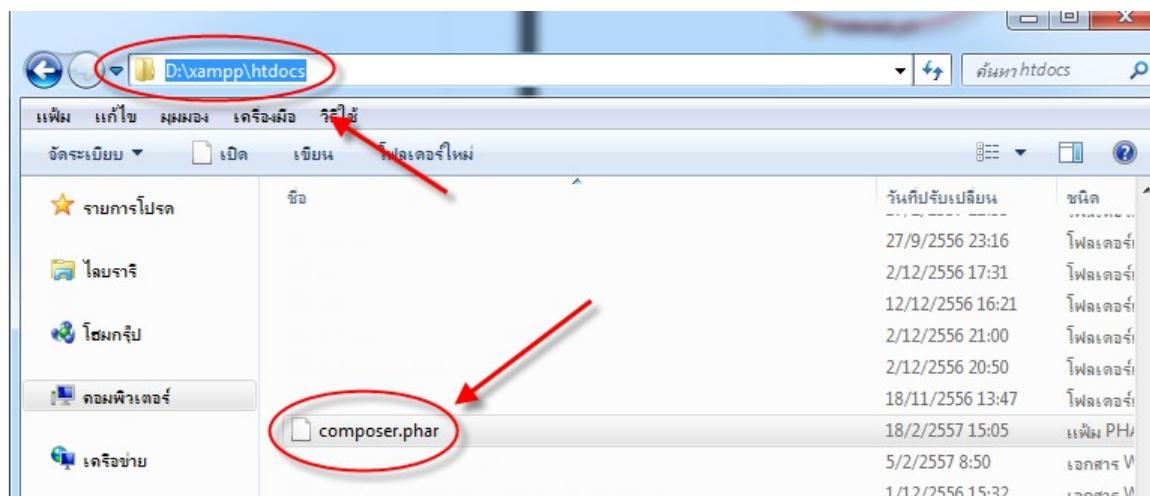
## • การติดตั้ง Laravel Framework

- สมมติ ผมต้องการสร้างชื่อ โปรเจกต์ tutorial\_laravel ฝากนั้น ให้เราไปสร้างโฟล์เดอร์ชื่อ tutorial\_laravel ใน server ของเรา (ตัวอย่าง ผมติดตั้ง XAMPP SERVER ผมจะสร้างโฟล์เดอร์ไว้ที่ D:\xampp\htdocs\tutorial\_laravel)



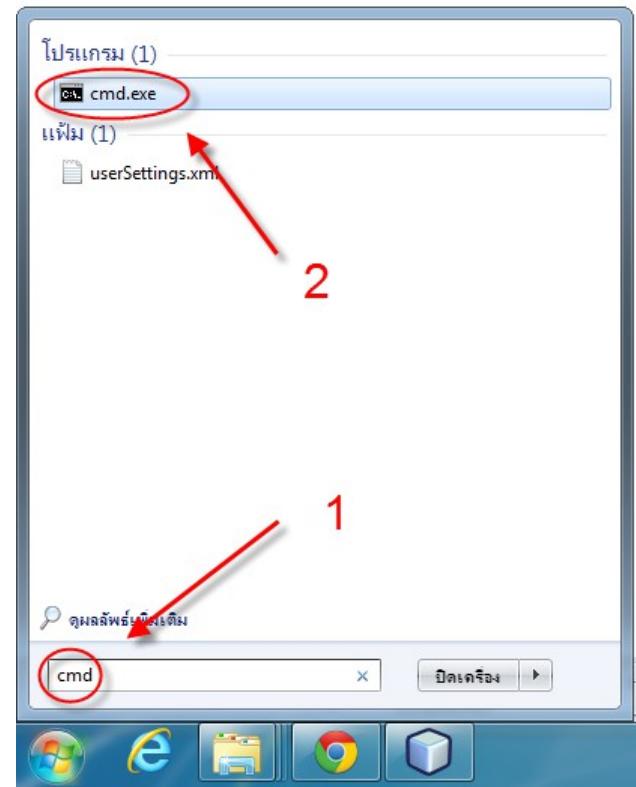
# บทที่ 5 ติดตั้ง Laravel

- การติดตั้ง Laravel Framework
  - ให้เรา Copyไฟล์ composer.phar ที่เราได้ดาวน์โหลดมา และนำไปวางไว้ในโฟเดอร์ C:\xampp\htdocs\



# บทที่ 5 ติดตั้ง Laravel

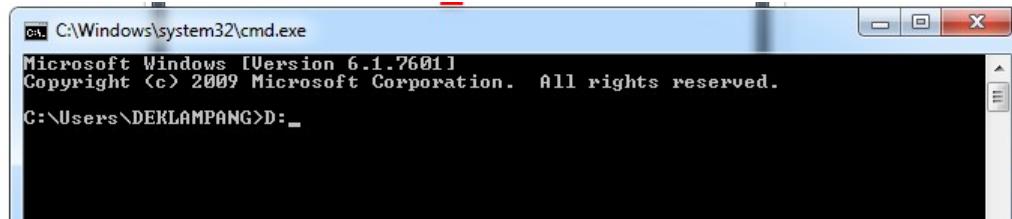
- การติดตั้ง Laravel Framework
  - เรียกใช้งาน Command line โดยให้เราเข้าไปที่ เมนู Start และ พิมพ์คำว่า "cmd" ในช่อง "ค้นหา" ดังภาพ
  - สุดท้ายให้คลิกที่ไฟล์ "cmd.exe"



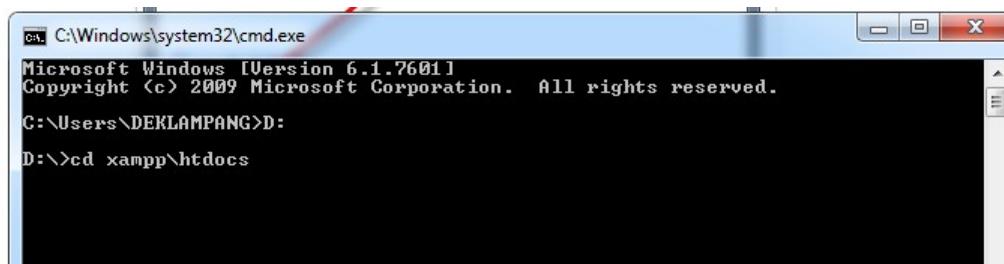
# บทที่ 5 ติดตั้ง Laravel

- การติดตั้ง Laravel Framework

- พิมพ์คำสั่ง "D:" และกดปุ่ม "Enter"

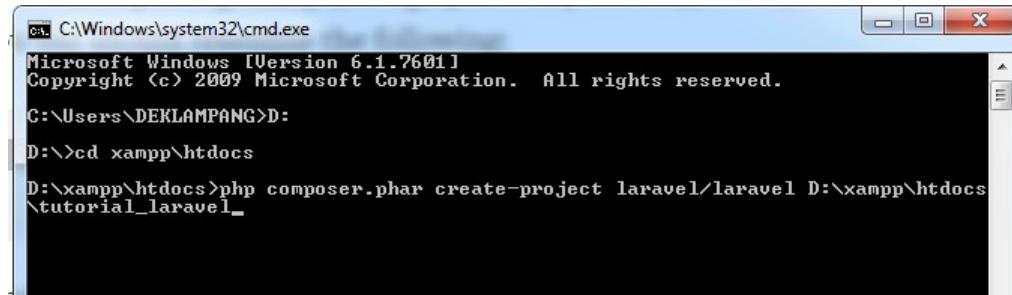


- พิมพ์คำสั่ง 'cd xampp\htdocs' และกดปุ่ม "Enter"



# บทที่ 5 ติดตั้ง Laravel

- การติดตั้ง Laravel Framework
  - พิมพ์คำสั่ง "php composer.phar create-project laravel/laravel D:\xampp\htdocs\tutorial\_laravel"
  - แลกดปุ่ม "Enter" แล้ว..รอสักครู่..

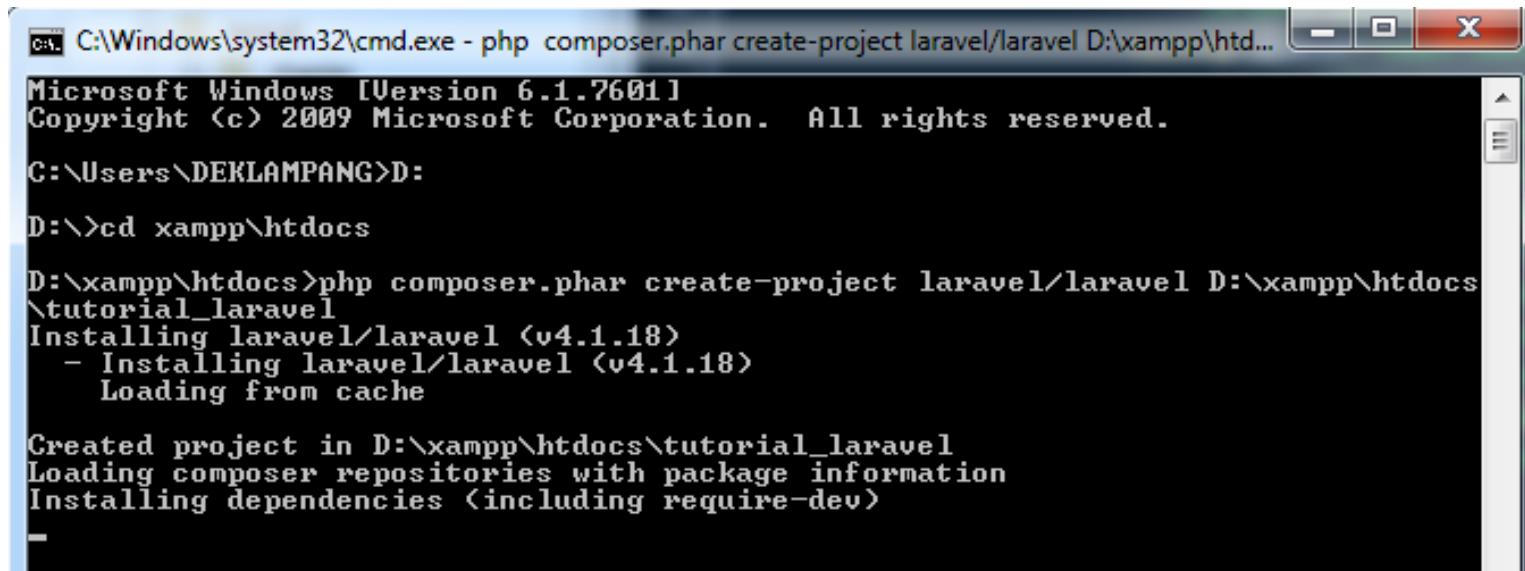


```
cmd C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\DEKLAMPANG>D:
D:>>cd xampp\htdocs
D:\xampp\htdocs>php composer.phar create-project laravel/laravel D:\xampp\htdocs\tutorial_laravel
```

# บทที่ 5 ติดตั้ง Laravel

- การติดตั้ง Laravel Framework
  - Composerเริ่มต้นการดาวโหลดไฟล์และติดตั้ง Laravel Framework ทั้งรูปภาพ



```
C:\Windows\system32\cmd.exe - php composer.phar create-project laravel/laravel D:\xampp\htdocs... Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\DEKLAMPANG>D:
D:>cd xampp\htdocs
D:\xampp\htdocs>php composer.phar create-project laravel/laravel D:\xampp\htdocs\tutorial_laravel
Installing laravel/laravel (v4.1.18)
- Installing laravel/laravel (v4.1.18)
  Loading from cache

Created project in D:\xampp\htdocs\tutorial_laravel
Loading composer repositories with package information
Installing dependencies (including require-dev)
-
```

# บทที่ 5 ติดตั้ง Laravel

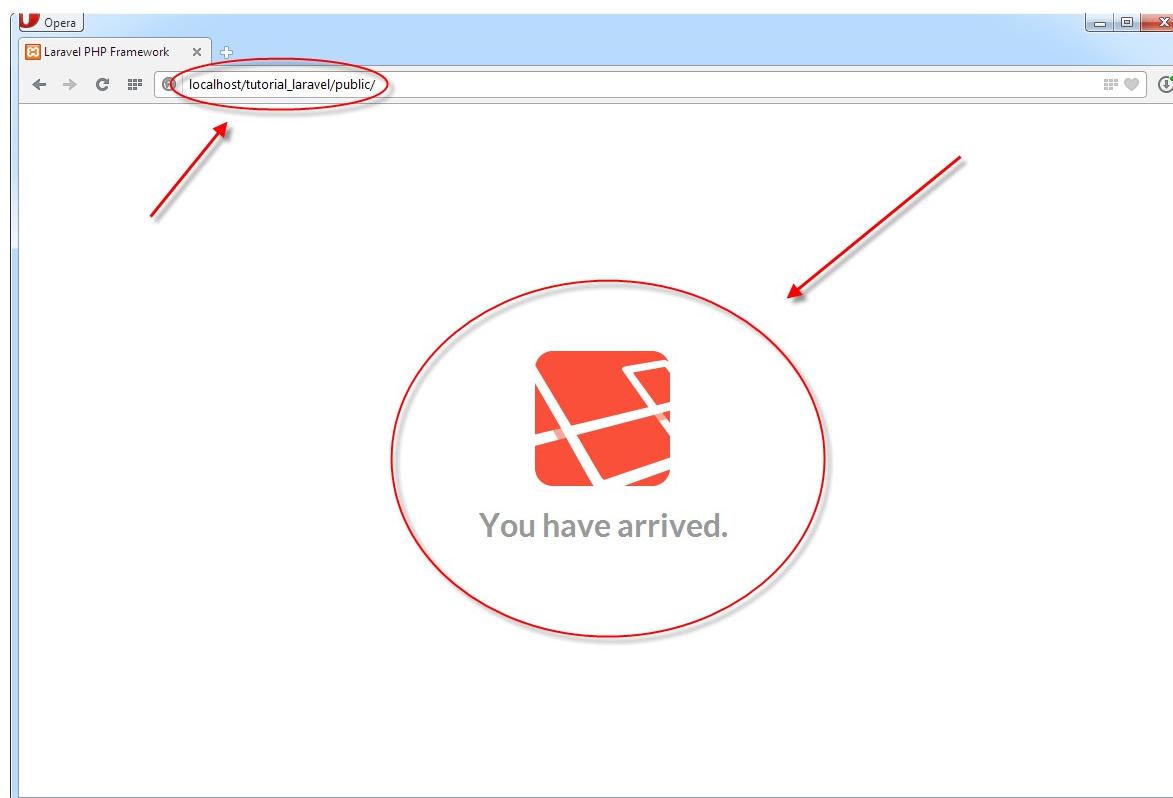
- การติดตั้ง Laravel Framework
  - รอให้ Composer ดาวน์โหลดไฟล์และติดตั้ง Laravel Framework สำเร็จ ดังรูปภาพ

```
C:\Windows\system32\cmd.exe
monolog/monolog suggests installing mlehner/gelf-php (Allow sending log messages
  to a GrayLog2 server)
monolog/monolog suggests installing raven/raven (Allow sending log messages to a
  Sentry server)
monolog/monolog suggests installing doctrine/couchdb (Allow sending log messages
  to a CouchDB server)
monolog/monolog suggests installing ruflin/elastica (Allow sending log messages
  to an Elastic Search server)
monolog/monolog suggests installing ext-amqp (Allow sending log messages to an A
MQP server (1.0+ required))
monolog/monolog suggests installing ext-mongo (Allow sending log messages to a M
ongoDB server)
monolog/monolog suggests installing aws/aws-sdk-php (Allow sending log messages
  to AWS services like DynamoDB)
d11wtq/boris suggests installing ext-readline (*)
d11wtq/boris suggests installing ext-pcntl (*)
d11wtq/boris suggests installing ext-posix (*)
laravel/framework suggests installing doctrine/dbal (Allow renaming columns and
dropping SQLite columns.)
Writing lock file
Generating autoload files
Generating optimized class loader
Application key [ue1Get6tR8X84uWESJ31tm0RwL8p15xS] set successfully.

D:\xampp\htdocs>
```

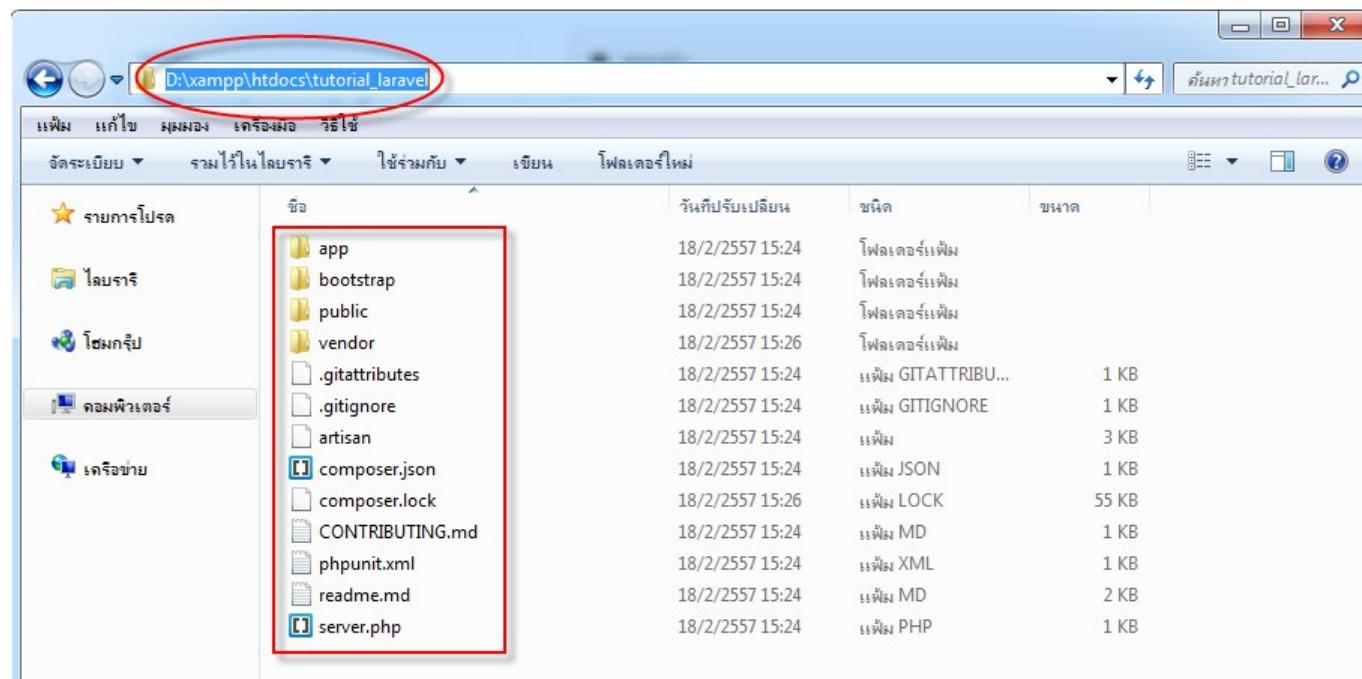
# บทที่ 5 ติดตั้ง Laravel

- การติดตั้ง Laravel Framework
  - ทดสอบ"การติดตั้ง Laravel Framework"
  - ให้เข้าไปที่`http://localhost/tutorial_laravel/public`แล้วติดตั้งสำเร็จจะเป็นดังรูปภาพ



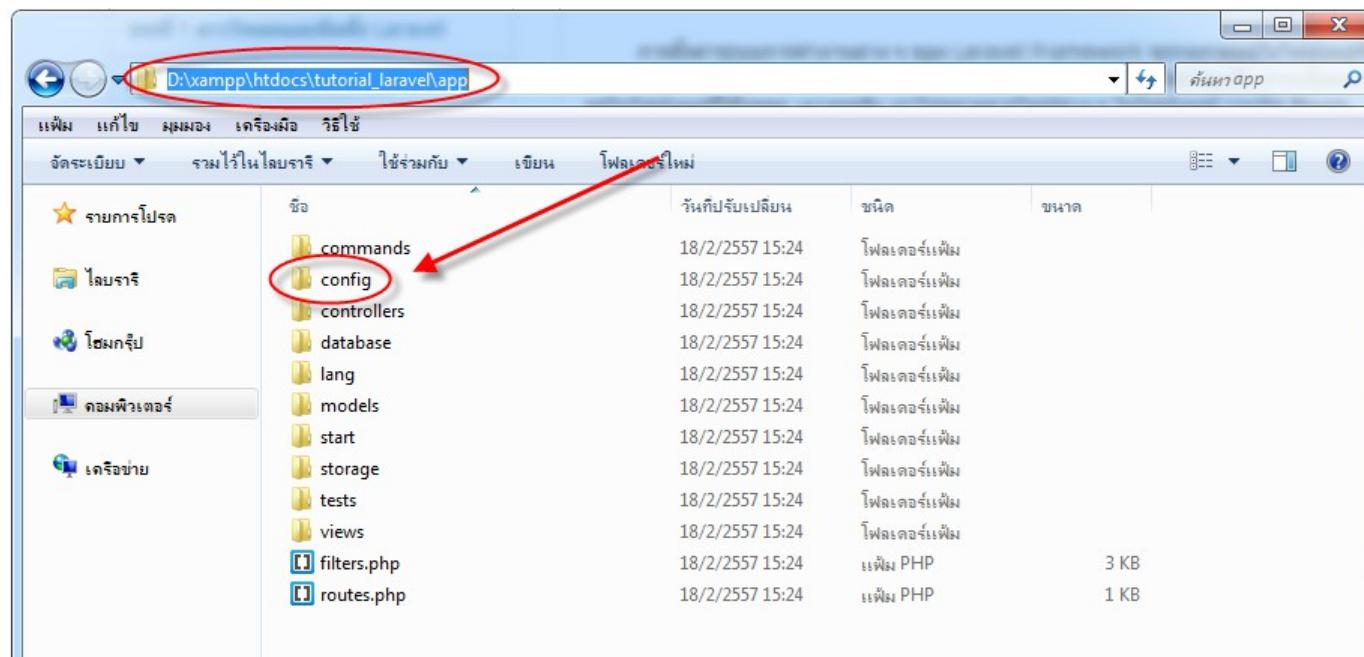
# บทที่ 6 เริ่มทำเว็บแรก

- โครงสร้าง Folder ของ โปรเจ็ค
  - โฟล์เดอร์และไฟล์ทั้งหมด ใน โฟล์เดอร์ "utorial\_laravel"



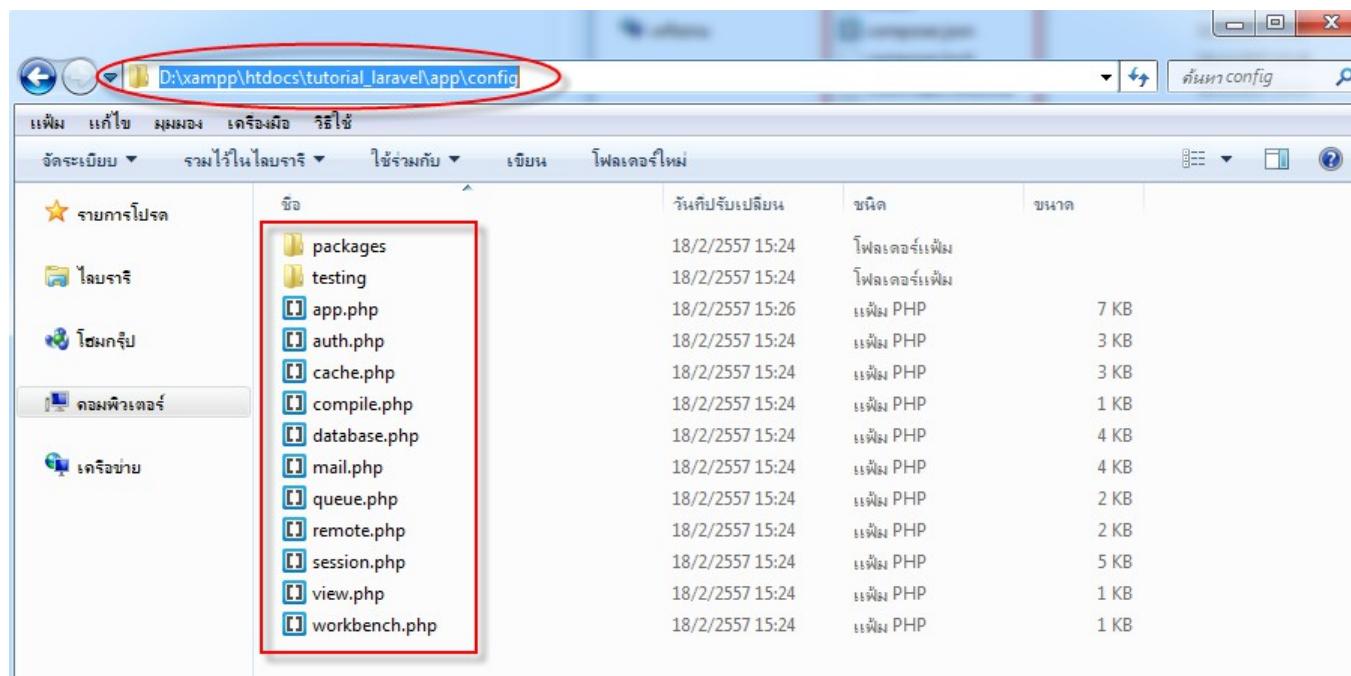
# บทที่ 6 เริ่มทำเว็บแรก

- โครงสร้าง Folder ของ โปรเจ็ค
  - โฟล์เดอร์ config จะอยู่ ในไฟล์เดอร์ app อีกชั้นหนึ่ง ครับ



# บทที่ 6 เริ่มทำเว็บแรก

- โครงสร้าง Folder ของ โปรเจ็ค
  - โฟล์เดอร์และไฟล์ต่าง ๆ ในโฟล์เดอร์ config"



# บทที่ 6 เริ่มทำเว็บแรก

- โครงสร้าง Folder ของ โปรเจ็ค
  - ทำความสะอาดรู้จักรไฟล์หลัก ๆ ที่เราควรรู้จัก
    - app.php :: กำหนดภาษาหลัก, สร้างชื่อย่อ (aliases) ให้กับ namespace และกำหนด URL หลัก เช่น `http://localhost`
    - auth.php :: กำหนดการรับรองความถูกต้อง ชื่อ Model, Database Table เป็นต้น
    - cache.php :: กำหนดรูปแบบการ Cache สนับสนุน Memcached และ Redis
    - database.php :: การหนดการเชื่อมต่อฐานข้อมูล ประเภทต่าง ๆ เช่น sqlite, mysql, pgsql, sqlsrv เป็นต้น
    - mail.php :: กำหนดค่า SMTP, HOST, PORT, USERNAME, PASSWORD และอื่น ๆ ให้กับ Email
    - session.php :: กำหนดค่า SESSION ต่าง ๆ อาทิเช่น ตั้งเวลาการเก็บค่า Session รูปแบบการจัดเก็บค่า Session เป็นต้น
    - view.php :: กำหนด Path ให้กับไฟล์ View ของเว็บแอพฯ ของเรา
    - workbench.php :: กำหนดค่าอื่น ๆ เช่น ชื่อของเรา, อีเมลของเรา เป็นต้น

# บทที่ 6 เริ่มทำเว็บแรก

- การตั้งค่าการทำงานของระบบหลัก
  - เข้าไปที่ไดเรกทอรี [skeleton]/app/config/app.php

```
1 | return array(
2 |   'debug' => true,
3 |   'url' => 'http://localhost',
4 |   'timezone' => 'UTC',
5 |   'locale' => 'en',
6 |   ....
```

- กำหนดค่าTimeZoneให้เป็น"Asia/Bangkok"

```
1 | return array(
2 |   ....
3 |   'timezone' => 'Asia/Bangkok',
4 |   ....
```

# บทที่ 6 เริ่มทำเว็บแรก

- ตั้งค่าการเชื่อมต่อฐานข้อมูล MySQL
  - เข้าไปที่ไฟล์ config/database.php

```
1 | return array(
2 |     'fetch' => PDO::FETCH_CLASS,
3 |     'default' => 'mysql',
4 |     'connections' => array(
5 |         'sqlite' => array(
6 |             'driver' => 'sqlite',
7 |             'database' => __DIR__.'../database/production.sqlite',
8 |             'prefix' => '',
9 |         ),
10 |
11         'mysql' => array(
12             'driver' => 'mysql',
13             'host' => 'localhost',
14             'database' => 'database',
15             'username' => 'root',
16             'password' => '',
17             'charset' => 'utf8',
18             'collation' => 'utf8_unicode_ci',
19             'prefix' => '',
20         ),
21         ....
```

# บทที่ 6 เริ่มทำเว็บแรก

- ตั้งค่าการเชื่อมต่อฐานข้อมูล MySQL
  - กำหนดค่าให้กับ อาร์เรย์ connections mysql

```
1 return array(
2     .....
3     'connections' => array(
4         .....
5         'mysql' => array(
6             'driver'      => 'mysql',
7             'host'        => 'localhost', // Host Database
8             'database'    => 'db_tutorial_laravel', // Name Database
9             'username'    => 'root', // Username Database
10            'password'   => '1234', // Username Database
11            'charset'    => 'utf8', // Charset Database
12            'collation'  => 'utf8_unicode_ci',
13            'prefix'     => '',
14        ),
15        .....

```

# บทที่ 6 เริ่มทำเว็บแรก

ออกแบบและสร้าง layout ด้วย Twitter Bootstrap v.3



The most popular front-end framework for developing responsive,  
mobile first projects on the web.

[Download Bootstrap](#)

Currently v3.1.1

# บทที่ 6 เริ่มทำเว็บแรก

- ติดตั้งไฟล์Twitter Bootstrap V.3.1.1
  - เข้าไปที่ URL ::  
<http://getbootstrap.com/getting-started/#download>

## Download

Bootstrap has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

### Bootstrap

Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.

[Download Bootstrap](#)

### Source code

Source Less, JavaScript, and font files, along with our docs. **Requires a Less compiler and some setup.**

[Download source](#)

### Sass

Bootstrap ported from Less to Sass for easy inclusion in Rails, Compass, or Sass-only projects.

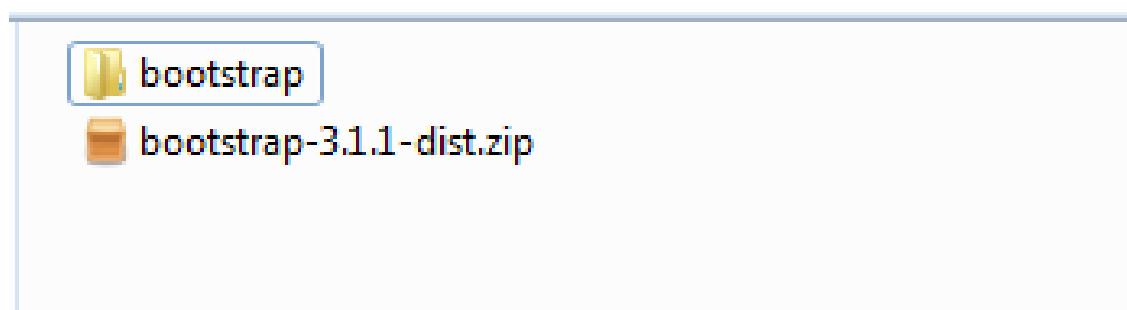
[Download Sass](#)

Currently v3.1.1.

# บทที่ 6 เริ่มทำเว็บแรก

- ติดตั้งไฟล์Twitter Bootstrap V.3.1.1

2. เข้าไปที่ไดเรคทอรีที่เก็บไฟล์ดาวโหลดไว้ และจะพบไฟล์ชื่อว่า "bootstrap-3.1.1-dist.zip" ของ Twitter Bootstrap ที่เราดาวน์โหลดมา และทำการแตกไฟล์ สุดท้ายเปลี่ยนไฟล์เดอร์จาก "bootstrap-3.1.1-dist" มาเป็น "bootstrap" แทน ตามรูปภาพ



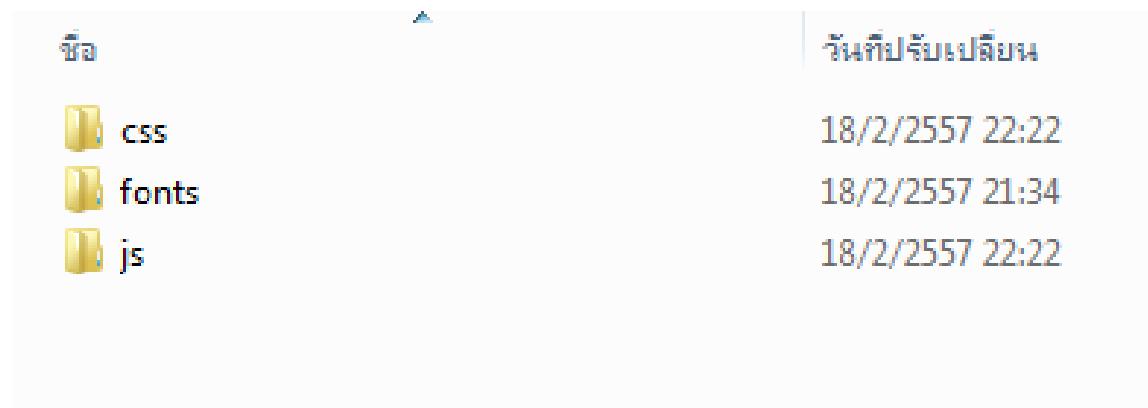
# บทที่ 6 เริ่มทำเว็บแรก

- ติดตั้งไฟล์Twitter Bootstrap V.3.1.1
  - 3. สร้างโฟล์เดอร์ชื่อว่า"assets"ในโฟล์เดอร์"public" และ Copy โฟล์เดอร์"bootstrap" (ตามข้อ2) เข้ามาไว้ใน โฟล์เดอร์"assets"



# บทที่ 6 เริ่มทำเว็บแรก

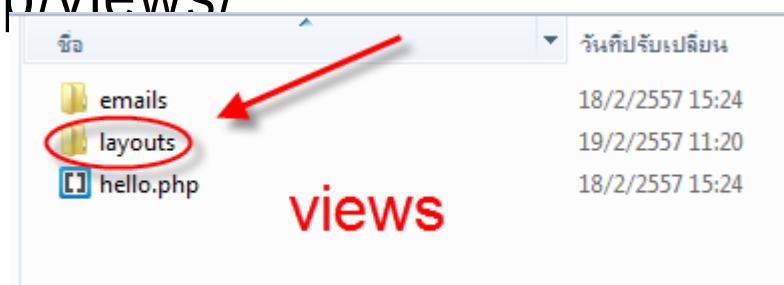
- ติดตั้งไฟล์Twitter Bootstrap V.3.1.1  
4. โฟล์เดอร์และไฟล์ต่าง ๆ ในโฟล์เดอร์"bootstrap"



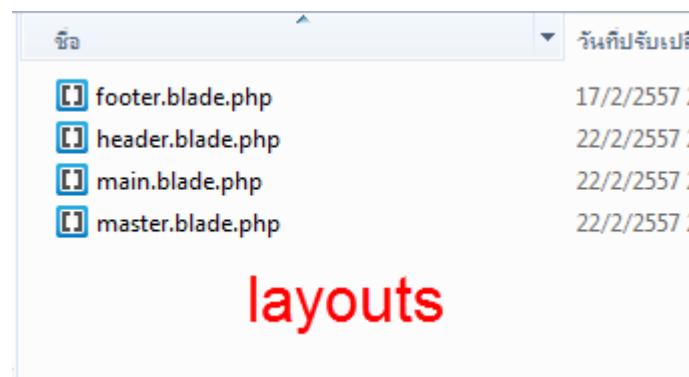
- 5. ติดตั้งไฟล์ Twitter Bootstrap V.3.1.1 สำเร็จ...แล้ว

# บทที่ 6 เริ่มทำเว็บแรก

- สร้าง Layouts ให้กับเว็บแอปพลิเคชันของเรา กันดีกว่า
  - ไฟล์ Layouts ทั้งหมดจะถูกเก็บไว้ในโฟล์เดอร์ "layouts" ให้เราสร้าง โฟล์เดอร์ "layouts" ขึ้นมาเอง ภายในไดเรคทอรี [skeleton]/app/views/



- ไฟล์ Layouts ทั้งหมด ในโฟล์เดอร์ "layouts"/



# บทที่ 6 เริ่มทำเว็บแรก

- สร้าง Layouts ให้กับเว็บแอปพลิเคชันของเรา กันดีกว่า

## 3. เขียนคำสั่งแสดงไฟล์ layouts เริ่มต้นไฟล์ "main.blade.php"

```
1  <!DOCTYPE html>
2  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3  <head>
4      
6      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
7      <meta name="language" content="en" />
8
9      <title> Tutorial Laravel 4 Framework </title>
10     <meta name="keywords" content="your, awesome, keywords, here" />
11     <meta name="author" content="Jon Doe" />
12     <meta name="viewport" content="width=device-width, initial-scale=1">
13     <meta name="description" content="Lorem ipsum dolor sit amet" />
14
15     
17     <meta name="viewport" content="width=device-width, initial-scale=1">
18
19     
21     <link rel="stylesheet" type="text/css" href="assets/bootstrap/css/bootstrap.min.css" />
22     <link rel="stylesheet" type="text/css" href="assets/bootstrap/css/jquery-ui-bootstrap.css" />
23     <script type="text/javascript" src="assets/bootstrap/js/jquery.js"></script>
24     <script type="text/javascript" src="assets/bootstrap/js/jquery-ui.min.js"></script>
25     <script type="text/javascript" src="assets/bootstrap/js/bootstrap.min.js"></script>
26 </head>
27 <body>
28     <!-- นำเข้าไฟล์ header.blade.php --&gt;
29     @include("layouts.header")
30     &lt;div class="container"&gt;
31         @yield("content")
32     &lt;/div&gt;
33     <!-- นำเข้าไฟล์ footer.blade.php --&gt;
34     @include("layouts.footer")
35 &lt;/body&gt;
36 &lt;/html&gt;</pre>
```

# บทที่ 6 เริ่มทำเว็บแรก

- สร้าง Layouts ให้กับเว็บแอปพลิเคชันของเรา กันดีกว่า
  - เขียนคำสั่งแสดงไฟล์ layouts เริ่มต้นไฟล์ "master.blade.php"

```
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
3 <head>
4     <!-- Basic Page Needs
5     ===== -->
6     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
7     <meta name="language" content="en" />
8
9     <title> Tutorial Laravel 4 Framework </title>
10    <meta name="keywords" content="your, awesome, keywords, here" />
11    <meta name="author" content="Jon Doe" />
12    <meta name="viewport" content="width=device-width, initial-scale=1">
13    <meta name="description" content="Lorem ipsum dolor sit amet" />
14
15    <!-- Mobile Specific Metas
16    ===== -->
17    <meta name="viewport" content="width=device-width, initial-scale=1">
18
19    <!-- CSS Bootstrap core CSS
20    ===== -->
21    <link rel="stylesheet" type="text/css" href="../assets/bootstrap/css/bootstrap.min.css" />
22    <link rel="stylesheet" type="text/css" href="../assets/bootstrap/css/jquery-ui-bootstrap.css" />
23    <script type="text/javascript" src="../assets/bootstrap/js/jquery.js"></script>
24    <script type="text/javascript" src="../assets/bootstrap/js/jquery-ui.min.js"></script>
25    <script type="text/javascript" src="../assets/bootstrap/js/bootstrap.min.js"></script>
26 </head>
27 <body>
28     <!-- นำเข้าไฟล์ header.blade.php -->
29     @include("layouts.header")
30     <div class="container">
31         @yield("content")
32     </div>
33     <!-- นำเข้าไฟล์ footer.blade.php -->
34     @include("layouts.footer")
35 </body>
36 </html>
```

# บทที่ 6 เริ่มทำเว็บแรก

- สร้าง Layouts ให้กับเว็บแอปพลิเคชันเรากันดีกว่า  
5. ต่อด้วยไฟล์ "header.blade.php"

```
1 @section("header")
2 <div class="container">
3   <div class="header">
4     <ul class="nav nav-pills pull-right">
5       <li class="active"><a href="#">Home</a></li>
6       <li><a href="#">About</a></li>
7       <li><a href="#">Contact</a></li>
8     </ul>
9     <h3 class="text-muted">Tutorial Laravel 4 Framework</h3>
10    </div>
11  </div>
12 @show
```

# บทที่ 6 เริ่มทำเว็บแรก

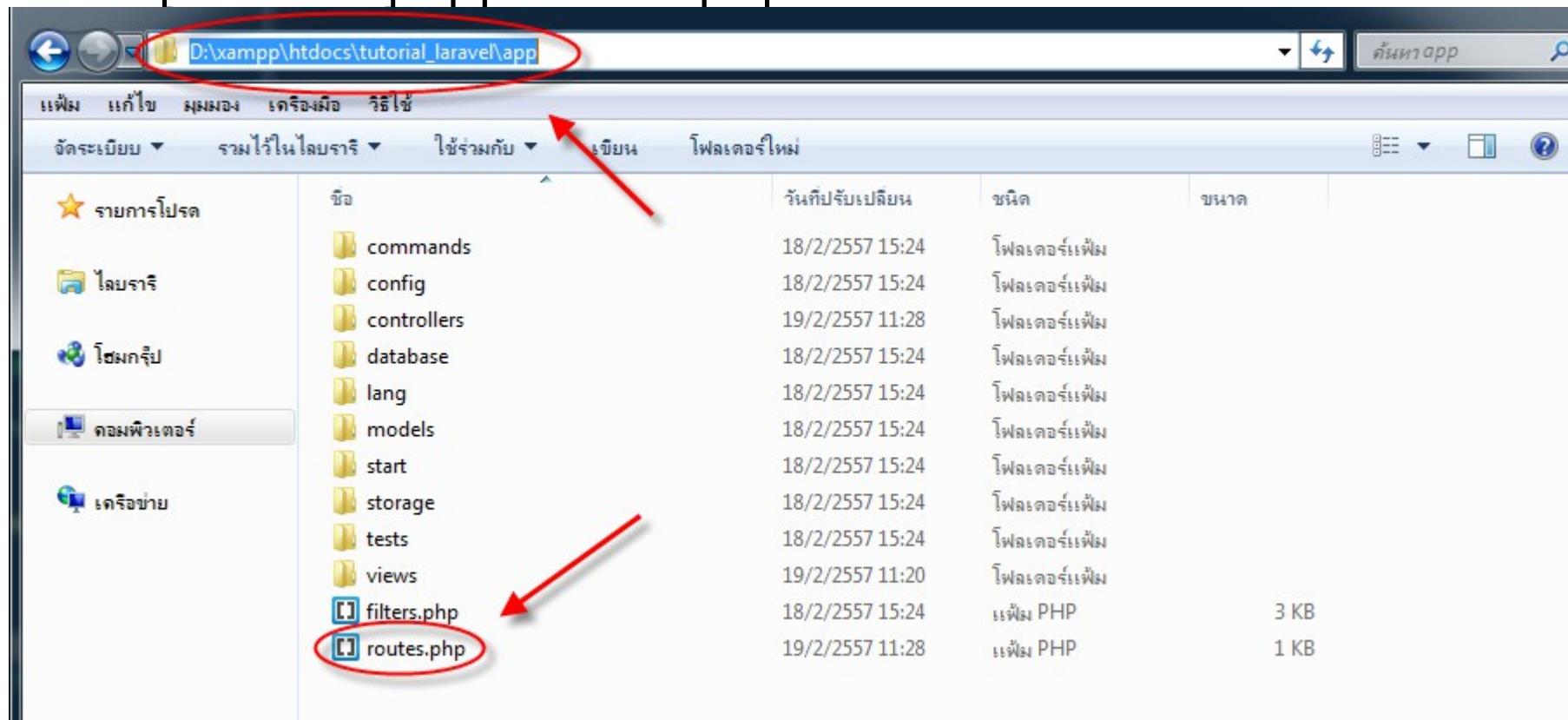
- สร้าง Layouts ให้กับเว็บแอปพลิเคชันของเรา กันดีกว่า  
6. สุดท้ายไฟล์ "footer.blade.php"

```
1 @section("footer")
2 <div class="footer">
3     <div class="container">
4         Powered by <a href="http://laravel.com/">Laravel</a>
5     </div>
6 </div>
7 @show
```

7. ติดตั้ง "Twitter Bootstrap 3" ให้กับ WebApp สำเร็จ..

# บทที่ 6 เริ่มทำเว็บแรก

- ทดสอบการติดตั้ง Twitter Bootstrap 3
  - เข้าเราเข้าไปแก้ไขไฟล์ routes.php ในไดเรกทอรี: [keleton]/app/routes.php



# บทที่ 6 เริ่มทำเว็บแรก

- ทดสอบการติดตั้ง Twitter Bootstrap 3

- ทำการแก้ไขไฟล์ routes.php คำสั่ง Route::get

```
1 | Route::get('/', 'SiteController@indexAction');
```

- เข้าไปสร้างไฟล์ Controller ชื่อว่า "SiteController.php" ในไดเรกทอรี :: [skeleton]/app/controllers/SiteController.php

```
1 | class SiteController extends BaseController {  
2 |     public function indexAction(){  
3 |         return View::make('index');  
4 |     }  
5 | }
```

# บทที่ 6 เริ่มทำเว็บแรก

- ทดสอบการติดตั้ง Twitter Bootstrap 3

4. เข้าไปสร้างไฟล์ index.php ชื่อว่า "index.blade.php" ในไดเรกทอรี:  
[skeleton]/app/views/index.blade.php

```
1 @extends("layouts.main")
2 @section("content")
3 <div class="container">
4     <div class="row">
5         <div class="col-lg-12">
6             <div class="jumbotron">
7                 <h1>Tutorial Laravel 4 Framework</h1>
8                 <p class="lead">ยินดีต้อนรับสู่บทเรียน Laravel 4 Framework  

9                     เรียนรู้วิธีการใช้งาน Laravel 4 Framework เพื่อนำมาexploit เคชั่น </p>
10                <p><a class="btn btn-lg btn-success" href="#" role="button">ดูกันเลย...</a></p>
11            </div>
12        </div>
13    </div>
14 </div>
15
16 @stop
```

# บทที่ 6 เริ่มทำเว็บแรก

- ทดสอบการติดตั้งTwitter Bootstrap 3

5. ทดสอบการติดตั้งTwitter Bootstrap v.3 ข้าไปที่ URL ::  
[http://localhost/tutorial\\_laravel/](http://localhost/tutorial_laravel/)

Tutorial Laravel 4 Framework

[Home](#) [About](#) [Contact](#)

## Tutorial Laravel 4 Framework

ยินดีต้อนรับสู่บทเรียน Laravel 4 Framework เรียนรู้วิธีการใช้งาน Laravel 4 Framework เพื่อนำมาพัฒนาแอปพลิเคชัน

ลุยกันเลย...

Powered by [Laravel](#)

6. ทดสอบ สำเร็จแล้ว...

# บทที่ 7 Url Routing คืออะไร

- เดิมการเรียกดูเพจของเว็บไซต์เราจะเรียกจากที่อยู่หรือ url ของไฟล์โดยตรง
  - เช่น <http://localhost/products.php>
- การทำ Url Routing คือการกำหนด url ว่าจะให้ชี้ไปเพื่อไหน หรือแสดงอะไร โดยไม่จำเป็นต้องมีไฟล์เพื่อจัดการ
  - เช่น <http://localhost/products>

# บทที่ 7 Url Routing คืออะไร

- Basic Routing

ในการตั้งค่าเราจะไปที่ [keleton]/app/routes.php โดยกำหนดค่าดังนี้

- การรับค่าที่เป็น get

```
1 Route::get('/', function()
2 {
3     return 'Hello World';
4 });
```

- การรับค่าที่เป็น POST

```
1 Route::post('foo/bar', function()
2 {
3     return 'Hello World';
4 });
```

# บทที่ 7 Url Routing คืออะไร

- Basic Routing

ในการตั้งค่าเราจะไปที่ [app/routes.php] โดยกำหนดค่าดังนี้

- การรับค่าได้หลายแบบ
  - เวอร์ชัน 4.1 ได้เพิ่มเมธอด match เพื่อช่วยให้เราไม่ต้องเขียน route get หรือ post สongtaw เพื่อรับค่าจาก route ซึ่งเดียวกัน

```
1 Route::match(array('GET', 'POST'), '/', function()
2 {
3     return 'Hello World';
4 });
```

- กำหนด route ในการเรียกพารามิเตอร์ foo ในทุกรูปแบบเมธอด

```
1 Route::any('foo', function()
2 {
3     return 'Hello World';
4 });
```

- ลิงก์ที่เรียกมาต้องเป็น https เท่านั้น

```
1 Route::get('foo', array('https', function()
2 {
3     return 'Must be over HTTPS';
4 }));
```

# บทที่ 7 Url Routing คืออะไร

- Route Parameters

ตัวอย่างการกำหนดรูปแบบของพารามิเตอร์

```
1 Route::get('user/{id}', function($id)
2 {
3     return 'User '.$id;
4 });
```

พารามิเตอร์แบบมีหรือไม่มี

```
1 Route::get('user/{name?}', function($name = null)
2 {
3     return $name;
4 });
```

กำหนดพารามิเตอร์แบบ

```
1 Route::get('user/{name?}', function($name = 'John')
2 {
3     return $name;
4 });
```

# บทที่ 7 Url Routing คืออะไร

- Route Parameters

การใช้ regex ตรวจสอบว่า parameter มีเทอร์รองกับที่กำหนดไว้ไหม

```
1 Route::get('user/{name}', function($name)
2 {
3     //
4 })
5 ->where('name', '[A-Za-z]+');
6
7 Route::get('user/{id}', function($id)
8 {
9     //
10 })
11 ->where('id', '[0-9]+');
```

สามารถใช้ array กำหนดขอบเขตของ parameter ที่ต้องการได้

```
1 Route::get('user/{id}/{name}', function($id, $name)
2 {
3     //
4 })
5 ->where(array('id' => '[0-9]+', 'name' => '[a-z]+'))
```

# บทที่ 7 Url Routing คืออะไร

- Route Filters

คือการกำหนดฟังก์ชันที่ใช้ในการตรวจสอบ หรือกำหนดเงื่อนไขในการถึง route ที่เราระบุ ซึ่งเราจะไปประยุกต์ใช้ใน app/filters.php

ตัวอย่างการสร้าง  
filter

```
1 Route::filter('old', function()
2 {
3     if (Input::get('age') < 200)
4     {
5         return Redirect::to('home');
6     }
7 });


```

การใส่ filter ให้  
route

```
1 Route::get('user', array('before' => 'old', function()
2 {
3     return 'You are over 200 years old!';
4 }));
```

# บทที่ 7 Url Routing คืออะไร

- Route Filters

การใช้ route หลายตัว

```
1 Route::get('user', array('before' => 'auth|old', function()
2 {
3     return 'You are authenticated and over 200 years old!';
4 }));
```

การกำหนดค่าเฉพาะให้ filter

```
1 Route::filter('age', function($route, $request, $value)
2 {
3     //
4 });
5
6 Route::get('user', array('before' => 'age:200', function()
7 {
8     return 'Hello World';
9 }));
```

# บทที่ 7 Url Routing คืออะไร

- Route Filters

filter บางตัวเรามีสิ่งให้ทำงานหลังจากที่ route ทำงานไปแล้วเราต้องกำหนดตัวแปร \$response เพื่อกำหนดค่าที่จะส่งไปให้ตัวฟังก์ชันด้วย

```
1 Route::filter('log', function($route, $request, $response, $value)
2 {
3     //
4 });


```

Pattern Based  
Filters

```
1 Route::filter('admin', function()
2 {
3     //
4 });
5
6 Route::when('admin/*', 'admin');
```

ตามตัวอย่างเรามีเพิ่ม filter ชื่อ admin เข้ากับทุกลิงค์ที่มี admin/ อยู่ข้างใน  
แล้วก็ยังสามารถกำหนดเมธอดให้ได้ด้วย

```
1 Route::when('admin/*', 'admin', array('post'));
```

# บทที่ 7 Url Routing คืออะไร

- Named Routes

คือการตั้งชื่อให้กับ route:

กำหนดให้ชื่อย่อนี้จะใช้ controller ไหน

ตอนนี้เราใช้ชื่อย่อ เพื่อสร้างลิงค์ได้แล้ว

เราใช้เมธอด currentRouteName เพื่อดึงชื่อของ Route ที่ทำงานในขณะนี้ได้

# บทที่ 7 Url Routing คืออะไร

- Route Groups

เราสามารถกำหนดกลุ่มให้ Route ได้ทำให้สัมภาระมากขึ้น

```
1 Route::group(array('before' => 'auth'), function()
2 {
3     Route::get('/', function()
4     {
5         // Has Auth Filter
6     });
7
8     Route::get('user/profile', function()
9     {
10        // Has Auth Filter
11    });
12});
```

# บทที่ 7 Url Routing คืออะไร

- Sub-Domain Routing

การสร้าง Route ให้กับโดเมนย่อย

```
1 Route::group(array('domain' => '{account}.myapp.com'), function()
2 {
3
4     Route::get('user/{id}', function($account, $id)
5     {
6         //
7     });
8
9 });


```

# บทที่ 7 Url Routing คืออะไร

- Route Prefixing

ในการกำหนดคำที่ใช้กำหนดกลุ่มของ เราใช้ prefix ในการตรวจสอบ

```
1 Route::group(array('prefix' => 'admin'), function()
2 {
3
4     Route::get('user', function()
5     {
6         //
7     });
8
9 });
```

# บทที่ 7 Url Routing คืออะไร

## • Route Model Binding

คือการผูกโมเดลเข้าไปกับ Route โดยใช้เมธอด Route::model

```
1 Route::model('user', 'User');
```

ต่อมาก็กำหนดให้มีการเรียกลิ้งที่มี {user} เป็นพารามิเตอร์

```
1 Route::get('profile/{user}', function(User $user)
2 {
3     //
4 });
```

เราจะทำการแทรก User instance เข้าไปใน Route ยกตัวอย่าง profile/1 ถูกเรียก User instance ก็จะมี ID = 1.

ถ้าพารามิเตอร์ที่ส่งเข้ามาไม่ตรงกับ model ได้ เราสามารถกำหนดการแสดงข้อผิดพลาดได้

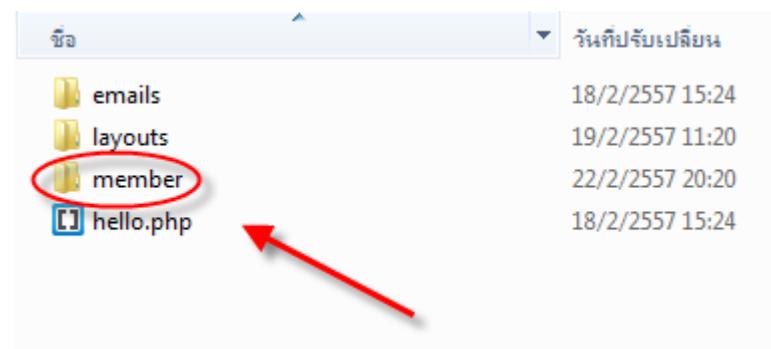
```
1 Route::model('user', 'User', function()
2 {
3     throw new NotFoundException;
4 });
```

# บทที่ 9 สร้าง View ไว้แสดงผล

- View คือส่วนกำหนดหน้าตาเว็บไซต์ โดยจะส่งผลลัพธ์เป็น HTML หรือ ข้อมูลตาม format ที่กำหนดกลับไปยัง web browser หรือ client
- Laravel 4 ใช้สคริปต์ช่วยสร้าง View template ที่เรียกว่า Blade Template
- ไฟล์ view template ทั้งหมด จะต้องมีนามสกุลเป็น .blade.php โดยทั้งหมดต้องอยู่ใน โฟลเดอร์ /app/views/
- การเขียน blade template จะมีทั้ง โค้ด HTML, PHP หรือ Blade เอง

# บทที่ 9 สร้าง View ไว้แสดงผล

- สมมติว่าเราจะสร้างเพจสำหรับจัดการข้อมูลสมาชิก หรือ Member
- สร้างไฟล์ View (index.blade.php)
  - สร้างโฟล์เดอร์ชื่อว่า "member" ไว้ในไดเรกทอรี [skeleton]/app/views/member



# บทที่ 9 สร้าง View ไว้แสดงผล

- สร้างไฟล์ View (index.blade.php)
  - สร้างไฟล์ชื่อว่า index.blade.php นี่ได้雷厨里:  
[skeleton]/app/views/member/index.blade.php

```
1  @extends("layouts.main") // ตั้ง Layout เข้ามาแสดงด้วย
2  @section("content")
3
4  <div class="container">
5    <div class="row">
6      <div class="col-lg-12">
7        <div class="jumbotron">
8          <h1>Tutorial Laravel 4 Framework</h1>
9          <p class="lead">ยินดีต้อนรับสู่เรียน Laravel 4 Framework เรียนรู้วิธีการใช้งาน Laravel 4 Framework เพื่อภาษาตัวเองและพัฒนา</p>
10         <p><a class="btn btn-lg btn-success" href="#" role="button">ดูเพิ่มเติม...</a></p>
11       </div>
12     </div>
13   </div>
14
15 <div class="container">
16   <div class="row">
17     <div class="col-lg-12">
18       <p class="alert alert-success">
19         บทที่ 4 ทดลองเรื่อง Model, View, Controller ตามโครงสร้าง PHP MVC </p>
20       <div class="col-lg-6">
21         <h4>Subheading</h4>
22         <p>Donec id elit non mi porta gravida at eget metus.</p>
23
24         <h4>Subheading</h4>
25         <p>Morbi leo risus, porta ac consectetur ac, vestibulum at eros.</p>
26
27         <h4>Subheading</h4>
28         <p>Maecenas sed diam eget risus varius blandit sit amet non magna.</p>
29     </div>
30
31     <div class="col-lg-6">
32       <h4>Subheading</h4>
33       <p>Donec id elit non mi porta gravida at eget metus. </p>
34
35       <h4>Subheading</h4>
36       <p>Morbi leo risus, porta ac consectetur ac, vestibulum at eros. </p>
37
38       <h4>Subheading</h4>
39       <p>Maecenas sed diam eget risus varius blandit sit amet non magna.</p>
40     </div>
41   </div>
42 </div>
43 </div>
44
45 @stop
```

# บทที่ 9 สร้าง View ไว้แสดงผล

- สร้างไฟล์Controller (MemberController.php)
  - สร้างไฟล์ชื่อว่า MemberController.php 之内 ไดเรกทอรี: [skeleton]/app/controller/MemberController.php

```
1 class MemberController extends BaseController {  
2     public function indexAction(){ // ชื่อ Action  
3         return View::make('member/index'); // member ชื่อโฟลเดอร์ และ index ชื่อไฟล์ index.blade.php  
4     }  
5 }  
6 }
```

# บทที่ 9 สร้าง View ไว้แสดงผล

- กำหนด route ให้ชี้ไปที่ controller member
  - กรณีเราสร้าง Controller ขึ้นมาทุกครั้ง ต้องมีการกำหนดค่า ให้กับ Routes ด้วยนะครับ ไฟล์อยู่ในไฟล์ routes.php ในโฟเดอร์ [skeleton]/app/routes.php

```
1 Route::any(
2     "member", // ชื่อที่กรอกใน URL
3     [
4         "as" => "member/index", // ชื่อ Controller และ Action ที่ต้องการรีบส่งเข้าไปหา
5         "uses" => "MemberController@indexAction" // ชื่อ Controller และ Action ที่ต้องการรีบส่งเข้าไปหา
6     ]
7 );
```

คำอธิบาย สมมุติผู้ใช้ กรอก URL "http://localhost/tutorial\_laravel/member" ระบบจะวิ่งไปหา memberController ในส่วนของ indexAction ที่เรากำหนดค่าต่าง ๆ ไว้

# บทที่ 9 สร้าง View ไว้แสดงผล

- ทดสอบการสร้าง Model, View, Controller แล้วนำเข้าไปที่ URL [http://localhost/tutorial\\_laravel/member](http://localhost/tutorial_laravel/member)

Tutorial Laravel 4 Framework

Home About Contact

## Tutorial Laravel 4 Framework

ยินดีต้อนรับสู่บทเรียน Laravel 4 Framework เรียนรู้วิธีการใช้งาน Laravel 4 Framework เพื่อนำพัฒนาแอปพลิเคชัน

ดูกันเลย...

บทที่ 4 ทดลองสร้าง Model, View, Controller ตามโครงสร้าง PHP MVC: คุณภาพการใช้งาน Laravel 4 Framework

### Subheading

Donec id elit non mi porta gravida at eget metus. Maecenas faucibus mollis interdum.

### Subheading

Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Cras mattis consectetur purus sit amet fermentum.

### Subheading

Maecenas sed diam eget risus varius blandit sit amet non magna.

### Subheading

Donec id elit non mi porta gravida at eget metus. Maecenas faucibus mollis interdum.

### Subheading

Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Cras mattis consectetur purus sit amet fermentum.

### Subheading

Maecenas sed diam eget risus varius blandit sit amet non magna.

# บทที่ 10 จัดการฐานข้อมูล

## • การตั้งค่าเบื้องต้น

- เราจะไปตั้งค่าที่ config/database.php ซึ่งเราจะไปตั้งค่าการเชื่อมต่อข้างในนี้ครับ
- โดย laravel ในขณะนี้สนับสนุน MySQL, Postgres, SQLite, and SQL Server.

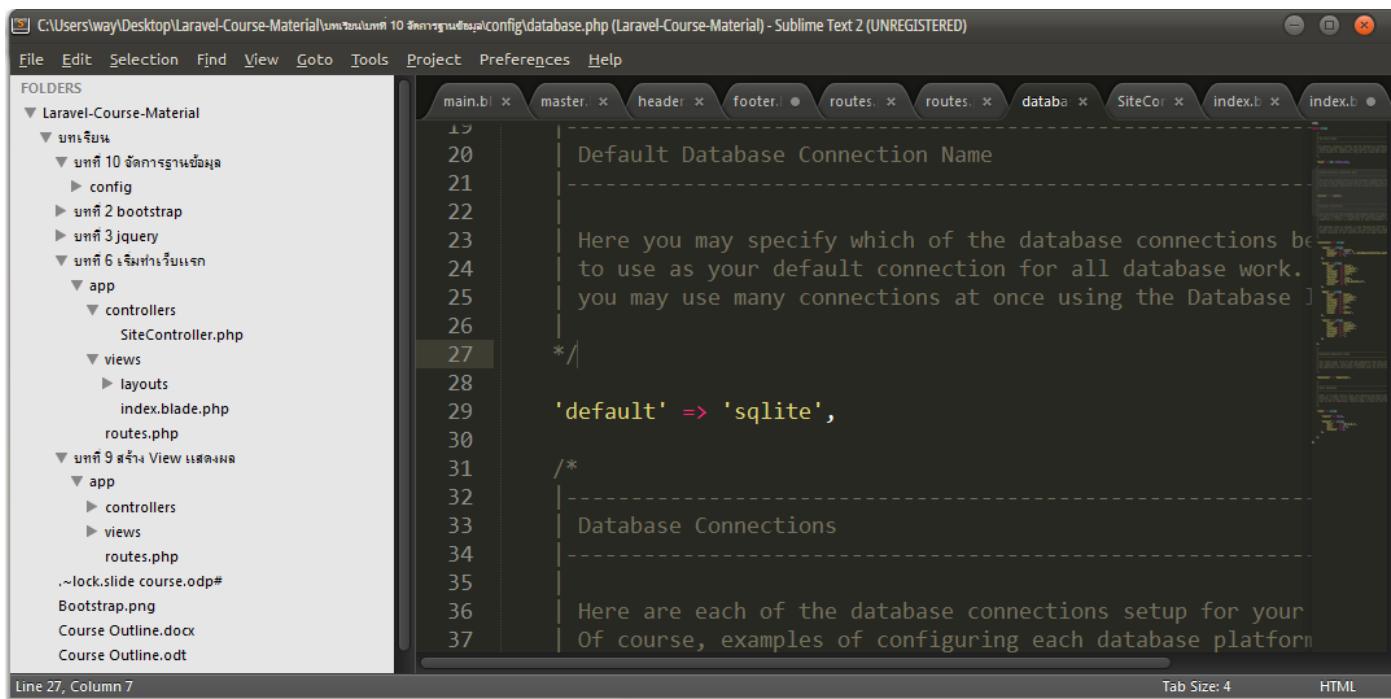
การกำหนดว่าจะให้ database ตัวไหนอ่านหรือ

**เขียน**  
ในตัวอย่างนี้ เราจะทำการโหลดбалานซ์ โดยแบ่งให้ database ตัวหนึ่งทำหน้าที่อ่านข้อมูล และอีกตัวหนึ่งทำหน้าที่เขียนข้อมูลอย่างเดียว

```
1 'mysql' => array(
2     'read' => array(
3         'host' => '192.168.1.1',
4     ),
5     'write' => array(
6         'host' => '196.168.1.2'
7     ),
8     'driver' => 'mysql',
9     'database' => 'database',
10    'username' => 'root',
11    'password' => '',
12    'charset' => 'utf8',
13    'collation' => 'utf8_unicode_ci',
14    'prefix' => '',
15 ),
```

# บทที่ 10 จัดการฐานข้อมูล

- การตั้งค่าเบื้องต้น
  - การเชื่อมต่อโดยใช้sqlite
  - ให้เราจะไปตั้งค่าที่ app/config/database.php
  - กำหนดค่า 'default' => 'sqlite',



The screenshot shows a Sublime Text 2 window displaying the contents of the `database.php` configuration file. The file is located in the `app/config` directory of a Laravel project named "Laravel-Course-Material". The code defines a default database connection and lists several database connection configurations.

```
C:\Users\way\Desktop\Laravel-Course-Material\บทที่ 10 จัดการฐานข้อมูล\config\database.php (Laravel-Course-Material) - Sublime Text 2 (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
└─ Laravel-Course-Material
    └─ บหที่ 10 จัดการฐานข้อมูล
        └─ config
            └─ database.php

main.bl header routes routes database SiteCor index.b index.b

19
20 | Default Database Connection Name
21 |
22 |
23 | Here you may specify which of the database connections be
24 | to use as your default connection for all database work.
25 | you may use many connections at once using the Database
26 |
27 */
28
29 'default' => 'sqlite',
30
31 /*
32 |
33 | Database Connections
34 |
35 |
36 | Here are each of the database connections setup for your
37 | Of course, examples of configuring each database platform
```

Line 27, Column 7      Tab Size: 4      HTML

# บทที่ 10 จัดการฐานข้อมูล

- สร้างตารางด้วย Schema Builder

- แนะนำให้สร้างตารางผ่าน Schema Builder เพราะเราสามารถนำไปใช้กับฐานข้อมูลทุกอย่างได้ทั้ง SQLite, MySQL, SQL Server, PostgreSQL
- คลาส Schema ใช้ในการจัดการตาราง มักใช้ร่วมกับคำสั่ง migration

สร้างและลบตาราง

สร้างตารางใหม่ด้วยเมธอด Schema::create ตามตัวอย่างเลขครับ

```
1 Schema::create('users', function($table)
2 {
3     $table->increments('id');
4});
```

# บทที่ 10 จัดการฐานข้อมูล

- เพิ่มคอลัมน์
  - ในการแก้ไขตารางเราใช้คำสั่ง Schema::table

```
1 Schema::table('users', function($table)
2 {
3     $table->string('email');
4 });
```

# บทที่ 10 จัดการฐานข้อมูล

- เพิ่มคอลัมน์
  - ในการแก้ไขตารางเราใช้คำสั่ง Schema::table

```
1 Schema::table('users', function($table)
2 {
3     $table->string('email');
4 });
```

# บทที่ 10 จัดการฐานข้อมูล

## • ชนิดของคอลัมน์

```
www.tuto.com  
01. $table->increments('id');  
02. ตั้ง auto increment ให้ id  
03. $table->string('email');  
04. ให้คอลั้น email เป็น varchar  
05. $table->string('name', 100);  
06. ให้คอลั้น name เป็น varchar มีความยาว 100  
07. $table->integer('votes');  
08. ให้คอลั้น votes เป็น integer  
09. $table->float('amount');  
10. ให้คอลั้น amount เป็น float  
11. $table->decimal('amount', 5, 2);  
12. ให้คอลั้น amount เป็น decimal ที่มีค่าระหว่าง 5 กับ 2  
13. $table->boolean('confirmed');  
14. ให้คอลั้น confirmed เป็น boolean  
15. $table->date('created_at');  
16. ให้คอลั้น created_at เป็น date  
17. $table->timestampe('added_on');  
18. ให้คอลั้น added_on เป็น timestamp  
19. $table->timestampe();  
20. ให้คอลั้น created_at กับ updated_at สร้างข้อมูลอัตโนมัติเวลาinsert  
21. $table->text('description');  
22. ให้คอลั้น description เป็น text  
23. $table->blob('data');  
24. ให้คอลั้น data เป็น blob
```

# บทที่ 10 จัดการฐานข้อมูล

- เพิ่ม Indexes

```
1 $table->string('email')->unique();
```

| COMMAND                                   | DESCRIPTION          |
|---|----------------------|
| \$table->primary('id');                   | เพิ่ม primary key    |
| \$table->primary(array('first', 'last')); | เพิ่ม composite keys |
| \$table->unique('email');                 | เพิ่ม unique index   |
| \$table->index('state');                  | เพิ่ม basic index    |

# บทที่ 10 จัดการฐานข้อมูล

- Foreign Key คืออะไร เชื่อม

```
1 $table->foreign('user_id')->references('id')->on('users');
```

ในตัวอย่างเราทำการให้คอลัมน์ `user_id` อ้างอิงกับคอลัมน์ `id` บนตาราง `users`

เราสามารถเสริมคำสั่ง "on delete" และ "on update" เข้าไปเพิ่มอีกด้วย

```
1 $table->foreign('user_id')
  2     ->references('id')->on('users')
  3     ->onDelete('cascade');
```

ในการลบใช้เมธอด `dropForeign` ตัวอย่าง

```
1 $table->dropForeign('posts_user_id_foreign');
```