

[HOME \(HTTP://WWW.W3PROGRAMMERS.COM/\)](http://WWW.W3PROGRAMMERS.COM/)
[CONTACT US \(HTTP://WWW.W3PROGRAMMERS.COM/CONTACT-US/\)](http://WWW.W3PROGRAMMERS.COM/CONTACT-US/)
[TRAININGS \(HTTP://WWW.W3PROGRAMMERS.COM/TRAININGS/\)](http://WWW.W3PROGRAMMERS.COM/TRAININGS/)
[CATEGORIES \(HTTP://WWW.W3PROGRAMMERS.COM/#\)](http://WWW.W3PROGRAMMERS.COM/#)
[বাংলা সংস্করণ \(HTTP://W3PROGRAMMERS.COM/BANGLA\)](http://W3PROGRAMMERS.COM/BANGLA)

PHP Professional এবং Zend Certified PHP Engineer (ZCPE ) Course এ সীমিত সংখ্যক আসন বাকি আছে। আগ্রহীদেরকে অতিসম্ভব মাসুদ আলম স্যার এর সাথে যোগাযোগ করার জন্য অনুরোধ করা যাচ্ছে। স্যার এর মোবাইল নম্বর : ০১৭২২ ৮১ ৭৫ ৯১

POSTED ON 1 MARCH, 2017 ([HTTP://WWW.W3PROGRAMMERS.COM/LARAVEL-ROUTE-GROUPS/](http://WWW.W3PROGRAMMERS.COM/LARAVEL-ROUTE-GROUPS/)) BY MASUD ALAM ([HTTP://WWW.W3PROGRAMMERS.COM/AUTHOR/MASUD1985/](http://WWW.W3PROGRAMMERS.COM/AUTHOR/MASUD1985/))

# Laravel 5.4 Tutorial Part-4: Laravel Route Groups

Laravel Route groups allow you to group several routes together, and apply any shared configuration settings once to the entire group, to reduce this duplication. Additionally, route groups are visual cues to future developers (and to your own brain) that these routes are grouped together and allowing you to keep your code neat and tidy. In reality, you're actually passing a closure to the group definition, and defining the grouped routes within that closure.

an example of a route group.

Example: A group of routes.

```

1 <?php
2
3 // routes/web.php
4 Route::group([], function () {
5
6     Route::get('/first', function () {
7         echo "I'm First";
8     });
9     Route::get('/second', function () {
10         echo "I'm Second";
11     });
12     Route::get('/third', function () {
13         echo "I'm Third";
14     });
15 });
16
17 );

```

## Sample Output

## Categories

PHP & MySQL Projects

(<http://www.w3programmers.com/category/php/php-mysql-projects/>)

PHP Basics

(<http://www.w3programmers.com/category/php-basics/>)

PHP Arrays, Strings and Numbers

(<http://www.w3programmers.com/category/php/php-arrays-strings-and-numbers/>)

PHP Control Structures

(<http://www.w3programmers.com/category/php-control-structures/>)

CODEIGNITER

(<http://www.w3programmers.com/category/codeigniter/>)

JQUERY and AJAX with PHP

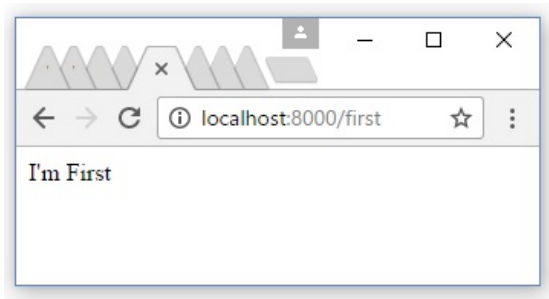
(<http://www.w3programmers.com/category/jquery-and-ajax-with-php/>)

JAVASCRIPT

(<http://www.w3programmers.com/category/javascript/>)

MySQL Tutorial

(<http://www.w3programmers.com/category/mysql-tutorial/>)



[\(http://www.w3programmers.com/basic-routing-and-view/route\\_group1/\)](http://www.w3programmers.com/basic-routing-and-view/route_group1/)

*Route Group*

We use the group() method, providing an array and closure. Inside the closure, we can define as many routes as we like!

By default, a route group doesn't actually do anything. There's no difference between the group and separating a segment of your routes with code comments. The empty array that's the first parameter, however, allows you to pass a variety of configuration settings that will apply to the entire route group.

## Path Prefixes

If many of your routes share a common URL structure, you could use a route prefix to avoid a small amount of repetition. Take a look at the following example:

```
1 <?php
2
3 // routes/web.php
4 Route::group(['prefix' => 'books'], function () {
5 // First Route
6 Route::get('/first', function () {
7 return 'The Colour of Magic';
8 });
9 // Second Route
10 Route::get('/second', function () {
11 return 'Reaper Man';
12 });
13
14 // Third Route
15 Route::get('/third', function () {
16 return 'Lords and Ladies';
17 });
18
19 });
```

Using the prefix array option of the route group, we can specify a prefix for all of the URLs defined within the group. For example, the three routes above are now accessible at the following URLs.

Example: The prefixed URLs.

/books/first

/books/second

/books/third

Use route prefixes to avoid repetition within your routes, and to group them by purpose for organizational or structural value.

WordPress

(<http://www.w3programmers.com/category/basic-wordpress-tutorial/>)

PHP Object Oriented Programming

(<http://www.w3programmers.com/category/php-object-oriented-programming/>)

PHP Security and Exceptions

(<http://www.w3programmers.com/category/php-security-and-exceptions/>)

PHP PDO, MySQLi and SQLite

(<http://www.w3programmers.com/category/php-pdo-mysqli-and-sqlite/>)

PHP Date Time and RegEx

(<http://www.w3programmers.com/category/php-date-time-and-regex/>)

PHP File, Mail, Session and Cookie

(<http://www.w3programmers.com/category/php-file-mail-session-and-cookie/>)

Useful PHP Functions and Features

(<http://www.w3programmers.com/category/useful-php-functions-and-features/>)

PHP Design patterns

(<http://www.w3programmers.com/category/php-design-patterns/>)

JSON, XML and Web Services

(<http://www.w3programmers.com/category/json-xml-and-web-services/>)

Standard PHP Library (SPL)

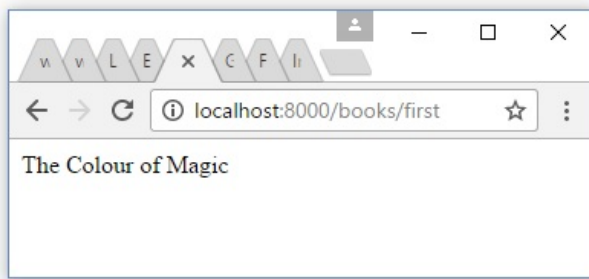
(<http://www.w3programmers.com/category/standard-php-library-spl/>)

Zend Framework 2.x

(<http://www.w3programmers.com/category/zend-framework-2-x/>)

Laravel

(<http://www.w3programmers.com/category/laravel/>)



[\(http://www.w3programmers.com/basic-routing-and-view/route-group-with-prefix/\)](http://www.w3programmers.com/basic-routing-and-view/route-group-with-prefix/)

*Route Group with Prefix*

## Subdomain Routing

Laravel Subdomain routing is the same as route prefixing, but it's scoped by subdomain instead of route prefix. If we use modern applications eg. wordpress, shopify, usually after signup we have our own subdomain for example masud.wordpress.com, masud.shopify.com, etc. In laravel, we can create features using Routing subdomain. Syntax essentially as follows:

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

For example, we create a website fakebook.dev in our local server, and it receive a sub-domain with username.

If we access masud.fakebook.dev it will display the page account Masud

If we access masud.fakebook.dev/profile it will display the profile Masud

If we access masud.fakebook.dev/status/1 it will display the status of Masud with id 1

## Step 1: Create virtual host and sub-domain

First we create a virtual host called fakebook.dev in our localhost host, that displays the user account information. To make it we need prepare your `fakebook.dev`, because the local host is not allowed to create a sub-domain, we need to add a domain for each user manually.

Let us create a virtual host called fakebook.dev and sub-domain called "masud" we add the following line in our hosts file

file: C: \ Windows \ System32 \ Drivers \ etc \ hosts

```
1 127.0.0.1 localhost
2 127.0.0.1 fakebook.dev
3 127.0.0.1 masud.fakebook.dev
4 127.0.0.1 sohel.fakebook.dev
5 ::1 localhost
```

## Step 2: Configure Apache :

Now Edit the Apache configuration file to accept Virtual Hosts and define the particular Virtual Hosts you want to setup on your computer.

File Path : /apache/conf/extra/httpd-vhosts.conf

Add Following code in httpd-vhosts.conf file

FAT FREE FRAMEWORK

(<http://www.w3programmers.com/category/fat-free-framework/>)

Facebook

(<http://www.w3programmers.com/category/facebook1/>)

Joomla

(<http://www.w3programmers.com/category/joomla/>)

Twitter Bootstrap

(<http://www.w3programmers.com/category/twitter-bootstrap/>)

Yii

(<http://www.w3programmers.com/category/yii/>)

SASS and LESS

(<http://www.w3programmers.com/category/sass-and-less/>)

Drupal

(<http://www.w3programmers.com/category/drupal/>)

HTML and CSS

(<http://www.w3programmers.com/category/html-and-css/>)

Angular JS

(<http://www.w3programmers.com/category/angular-js-2/>)

CakePHP

(<http://www.w3programmers.com/category/cakephp/>)

Android Application Development

(<http://www.w3programmers.com/category/android-apps-development/>)

Magento

(<http://www.w3programmers.com/category/magento/>)

Symfony

(<http://www.w3programmers.com/category/symfony/>)

Python

(<http://www.w3programmers.com/category/python/>)

```

1 <VirtualHost *:80>
2 DocumentRoot "C:\xampp\htdocs\blog\public"
3 ServerName fakebook.dev
4 <directory "C:\xampp\htdocs\blog\public">
5     Options Indexes FollowSymLinks
6     AllowOverride all
7     Order Deny,Allow
8     Deny from all
9     Allow from all
10 </directory>
11 </VirtualHost>

```

## Step 2: Add Subdomain Routing Code

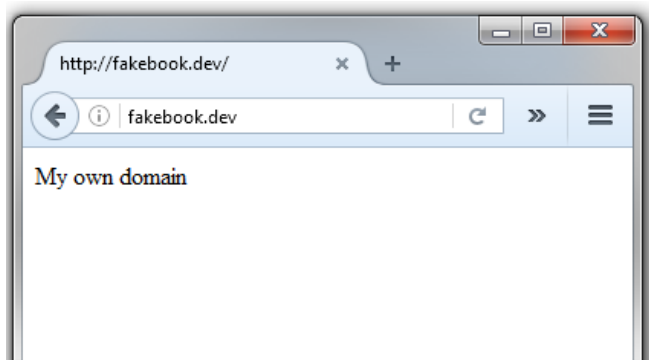
Now Add following code in your routes/web.php file:

```

1 Route::group(['domain' => 'fakebook.dev'], function()
2 {
3     Route::any('/', function()
4     {
5         return 'My own domain';
6     });
7 });
8
9 Route::group(['domain' => '{username}.fakebook.dev'], function()
10 {
11     Route::any('/', function($username)
12     {
13         return 'You visit your account: '. $username;
14     });
15 });

```

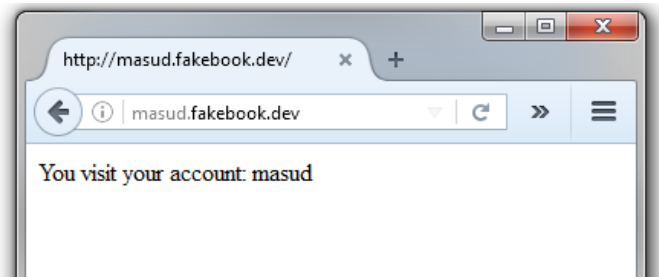
Once ready, Restart your xampp apache server and let's check your virtual host with Laravel:



<http://www.w3programmers.com/build-subdomain-routing-with-laravel/laravel-virtual-host-example/>

*Laravel Virtual Host Example*

And Now Check Your Subdomain Example:



<http://www.w3programmers.com/build-subdomain-routing-with-laravel/laravel-subdomain-routing-example-2/>

*Laravel Subdomain Routing Example*

## Step 3: Display User Profile with Laravel Subdomain and Group Routing Feature

Node.js

(<http://www.w3programmers.com/category/node-js/>)

PHP Exercises

(<http://www.w3programmers.com/category/php-exercises/>)

Bangla Article

(<http://www.w3programmers.com/category/bangla-article/>)

Join Our Group



**W3programmers**

Facebook Group · 6,155 members

[Join Group](#)

([https://www.facebook.com/plugins/group/jcgroup\\_id=307073372716694&source=email\\_ca](https://www.facebook.com/plugins/group/jcgroup_id=307073372716694&source=email_ca))

w3programmers is the Official group of the www.w3prog website. Here you can discuss everything about Web D Technologies such...



**W3programmers**  
34K likes

[Like Page](#)

Be the first of your friends to like this



**W3programmers**  
about a week ago

SEPTEMBER 2018

S	S	M	T
1	2	3	4
8	9	10	11
15	16	17	18
22	23	24	25
29	30		

For simplicity, let's save all user data in the array. Then, we are passing this array to closure on route to display profile. So that the route turns into : routes/web.php file:

S

S

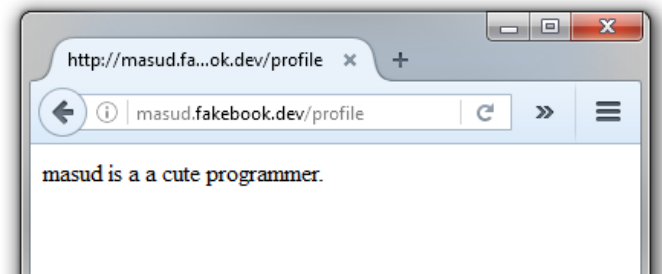
M

« Dec

(<http://www.w3programmers.com/2017/12/>)

```
1 Route::group(['domain' => 'fakebook.dev'], function()
2 {
3     Route::any('/', function()
4     {
5         return 'My own domain';
6     });
7 });
8
9 Route::group(['domain' => '{username}.fakebook.dev'], function()
10 {
11     Route::any('/', function($username)
12     {
13         return 'You visit your account: '. $username;
14     });
15
16     $data_user = [
17         'masud' => [
18             'profile' => ' a cute programmer. ',
19             'status' => [ 'I\'m cool!', 'I\'m cool very Cool!',
20 'Fantastic!']
21         ],
22         'sohel' => [
23             'profile' => 'a boss programmer.',
24             'status' => [ 'Sweet!', 'Today is incredible!', 'Nice ..']
25         ]
26     ];
27
28     Route::get('profile', function($username) use ($data_user)
29     {
30         return $username." is a ".$data_user[$username] ['profile'];
31     });
32 });
```

And Now Check Your User Subdomain and User Profile Example:



(<http://www.w3programmers.com/build-subdomain-routing-with-laravel/user-profile-with-subdomain-in-laravel-route/>)

*User Profile with Subdomain in Laravel Route*

## Step 4: Displaying User Status:

To display user status, we will also use the array that was created earlier. Add the following route:

```
1 Route::group(['domain' => 'fakebook.dev'], function()
2 {
3     Route::any('/', function()
4     {
5         return 'My own domain';
6     });
7 });
8
9 Route::group(['domain' => '{username}.fakebook.dev'], function()
10 {
11     Route::any('/', function($username)
12     {
13         return 'You visit your account: '. $username;
14     });
15
16     $data_user = [
17         'masud' => [
18             'profile' => ' a cute programmer. ',
19             'status' => [ 'I\'m cool!', 'I\'m cool very Cool!',
```

```

20     'status' => [ 'I\'m cool!', 'I\'m cool very Cool!',
21     'Fantastic!'],
22     ],
23     'sohel' => [
24         'profile' => 'a boss programmer.',
25         'status' => [ 'Sweet!', 'Today is incredible!', 'Nice ..' ]
26     ];
27
28
29     Route :: get ('status/{id}', function ($username, $id) use
30     ($data_user)
31     {
32         return $username. ' writes: '. $data_user [$username] ['status']
33         [$id];
34     });

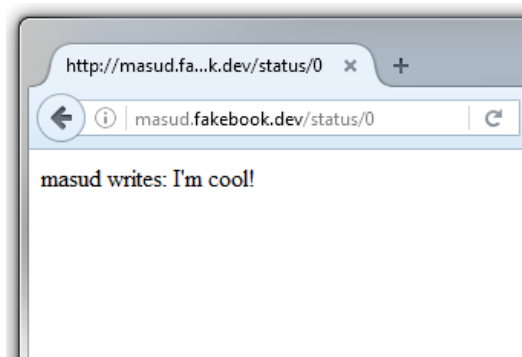
```

Let's try to check the status Masud:



(<http://www.w3programmers.com/build-subdomain-routing-with-laravel/user-status-with-subdomain-in-laravel-route/>)

*User Status with Subdomain in Laravel Route*



(<http://www.w3programmers.com/build-subdomain-routing-with-laravel/user-status-with-subdomain-in-laravel-routes/>)

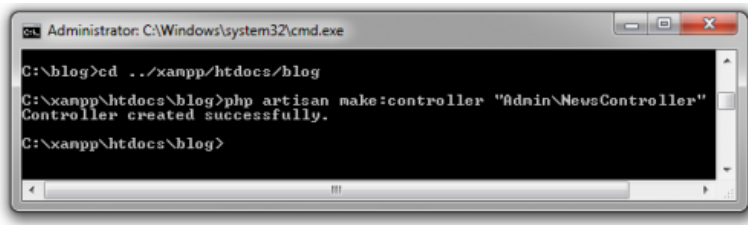
*User Status with Subdomain in Laravel Routes*

## Namespace Prefixes

Another common use-case for route groups is assigning the same PHP namespace to a group of controllers using the namespace parameter in the group array.

As an example let's use the namespace array attribute.

Say we have a controller called NewsController that contains all the Admin logic for your app's news section. You may place this file within the 'App/Http/Controllers/Admin' directory.



[\(http://www.w3programmers.com/laravel-route-groups/create-controller/\)](http://www.w3programmers.com/laravel-route-groups/create-controller/)

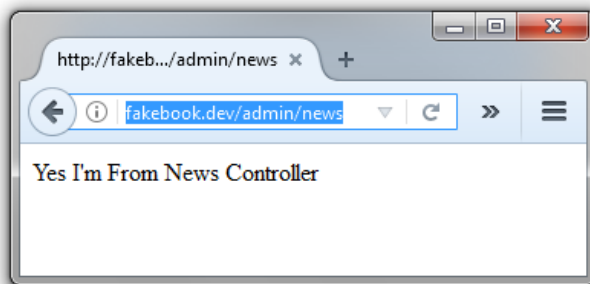
*Create Laravel Controller in a folder*

Laravel 5X follows [PSR-4 autoloading](http://www.php-fig.org/psr/psr-4/) (<http://www.php-fig.org/psr/psr-4/>) conventions, so the application expects the namespace to be identical to the path of the file, so our class might look something like this:

```
1 <?php
2
3 namespace App\Http\Controllers\Admin;
4
5 use Illuminate\Http\Request;
6 use App\Http\Controllers\Controller;
7
8 class NewsController extends Controller
9 {
10     //
11 }
```

We could write a route to this class like so:

```
1 Route::get('admin/news', [
2     'uses' => 'Admin\NewsController@index'
3 ]);
```



[\(http://www.w3programmers.com/laravel-route-groups/controller-output/\)](http://www.w3programmers.com/laravel-route-groups/controller-output/)

*Controller Output*

*Note: Laravel automatically assumes all your controllers will be in the App/Http/Controllers directory so we can leave this out of any controller declarations in the routes file.*

The above should work fine, but maybe you also have a dozen or so other class files that deal with Admin logic all within the same namespace. We can use the namespace option to group these together.

```
1 Route::group(['namespace' => 'Admin'], function()
2 {
3     Route::get('admin/news', [
4         'uses' => 'NewsController@index'
5     ]);
6
7     Route::get('admin/users', [
8         'uses' => 'UserController@index'
9     ]);
10 }
```

```
11 }));
```

Notice how I no longer define the Admin namespace for the controller for each route.

The same process can be applied to middleware, subdomains, and url prefixes.

## Name Prefixes

Lets take the first example and build on it. As you can see from the above route declarations all our admin routes share a common url prefix.

```
http://fakebook.dev/admin/news
http://fakebook.dev/admin/users
```

We can use the prefix array attribute to define the common url for our routes. In our case this is admin.

Our updated Route declarations would look like so.

```
1 Route::group(['namespace' => 'Admin', 'prefix' => 'admin'], function()
2 {
3     Route::get('news', [
4         'uses' => 'NewsController@index'
5     ]);
6
7     Route::get('users', [
8         'uses' => 'UserController@index'
9     ]);
10
11     ...
12 }));
```

You probably wondering why would this be useful? Well imagine you have developed a large application with tens if not hundreds of routes. Then one day your boss comes to you and says “Hey Mr. tester, we need to change the admin url from /admin to /cms, how long will this take?”.

If you’ve declared all your routes using groups with the prefix array attribute like above, this is going to be an easy and painless process for you.



**Masud Alam**

(<http://www.w3programmers.com/author/masud1985/>)

Hi, My name is Masud Alam, love to work with Open Source Technologies, living in Dhaka, Bangladesh. I’m a Certified Engineer on ZEND PHP 5.3, I served my first five years a number of leadership positions at Winux Soft Ltd, SSL Wireless Ltd, Canadian International Development Agency (CIDA), World Vision, Care Bangladesh, Helen Keller, US AID and MAX Group where I worked on ERP software and web development., but now i’m a founder and CEO of TechBeeo Software Company Ltd. I’m also a Course Instructor of ZCPE PHP 7 Certification and professional web development course at w3programmers Training Institute – a leading Training Institute in the country.



## One comment on “Laravel 5.4 Tutorial Part-4: Laravel Route Groups”



HARISH

REPLY (#COMMENT-41591)

30 MAY, 2018 AT 8:25 PM ([HTTP://WWW.W3PROGRAMMERS.COM/LARAVEL-ROUTE-GROUPS/#COMMENT-41591](http://www.w3programmers.com/laravel-route-groups/#comment-41591))

Hi, nice tutorial.

Using route group we can combine large number of routes in group to share route attribute like namespaces or middleware.

I have published an article on same topic, i which all Laravel routing are explained. Check it : <https://www.codefixup.com/routing-in-laravel/>

Thanks

### Leave a Reply

YOUR EMAIL ADDRESS WILL NOT BE PUBLISHED. REQUIRED FIELDS ARE MARKED

COMMENT

NAME \*

EMAIL \*

POST COMMENT