

## Build Blog System Using Laravel 5.3

### **Setup Database(Create posts, comments, users table)**

<http://www.hc-kr.com/2016/11/how-to-build-blog-system-with-laravel-53.html>

1. we will create new laravel projet (myblog) using composer  
>> `composer create-project --prefer-dist laravel/laravel myblog`
2. setting up our database connection (.env)  
create connection, make sure you have create a database in your server
3. we will create default authentication from Laravel  
>> `php artisan make:auth`
4. create new migration  
>> `php artisan make:migration create_post_table`  
>> `php artisan make:migration create_comment_table`
5. we need to add new field in users migration that we have before  
`$table->enum('role', ['admin', 'author', 'subscriber'])->default('author');`
6. create posts table
7. create comments table
8. Finally, run migration  
>> `php artisan migrate`
9. I'll try to register and then login first time  
So, default user after registered as an author

Next lessons, we will showing data from database, create single page and comments system.

The author can create, edit, delete a posts.

### **Single Page, Add Comments**

Previews lessons, we have created table and migrations

1. we will create Posts and Comments Model  
>> `php artisan make:model Posts`  
>> `php artisan make:model Comments`  
then, write relationship in Model file
2. Create new controller  
>> `php artisan make:controller PostsController --resource`  
>> `php artisan make:controller CommentsController --resource`

6. we will insert data into database before

7. <http://localhost:8000/blog>

### **Insert Data, Validation, Redirect Page and Flash Message**

create form to create new data and save into database, validation

1. go to BlogController, write code at create()

2. create a view (blog/create.blade.php), write code

so, now we have new view for create new data into database

<http://localhost:8000/blog/create>

if you click post button, we will have error

so, we need a token, if we have a token, the error will be hidden/ not displaying

3. make a validation to insert our new data (BlogController's store() write code)

4. we will show the error validation in create view, we can modife our validation like this ...

5. now, we will save the data. (BlogController's store() write code)

<http://localhost:8000/blog/create>

6. we need a redirect page after save the data.

and, in blog/index.blade.php file, we must add session message.

```
{{Session::get('message')}}}
```

<http://localhost:8000/blog/create>

### **Single page, Error 404 Page, Link to Single Page Laravel 5.3**

1. create single page of our title, that mean single page (detail page / article).

at BlogController.php show() write code

2. create new file in our view(blog/detail.blade.php write code)

<http://localhost:8000/blog/1>

3. if the article id is not found, we need to redirect it to 404 page not found.

at BlogController.php show() write code,

and create new file write code (views/errors/404.blade.php)

<http://localhost:8000/blog/400>

4. we need at blog/index.blade.php page, title as a link. the title will linkto a single page.

<http://localhost:8000/blog/>



### 3. show all posts

- PostsController.php index() write code
- create new routes
- create new views

<https://github.com/BlackrockDigital/startbootstrap-clean-blog>

a. Download bootstrap template for blog here

<https://startbootstrap.com/template-overviews/clean-blog/>

After download bootstrap templates, just copy all file to laravel Project (Public Folder) see on videos tutorial.

b. Replace app.blade.php with our bootstrap theme

(從下載的資料夾中的index.html複製全部去修改code)

<http://localhost:8000/>

c. Showing the post in home.blade.php page (write code)

then, 到DB posts資料表輸入測試資料

<http://localhost:8000/>

### 4. We need new function to show the single post

- PostsController.php show(\$slug) write code
- create new views
  - a. Create new folder (posts) in resources/Views directory,
  - b. Create show.blade.php in resources/Views/posts
  - c. Show the single post in show.blade.php page

- create new routes

Route::get('/{slug}', 'PostsController@show')->where('slug', '[A-Za-z0-9- \_]+');

<http://localhost:8000/my-next-post>

### 5. We need to show the Posts title as our header

- app.blade.php
- home.blade.php
- show.blade.php

<http://localhost:8000/my-next-post>

### 6. we need to change the Header background with our Post images

- 到DB幫post加上images的圖片路徑(用原本資料夾內的圖片即可)
- app.blade.php 修改圖片路徑

### 7. we will add login/register menu that mean if you're user login/register will showing, but if has logged your name will show on.

- app.blade.php write code

### 8. we need to add new master template (appadmin.blade.php)

- 修改login.blade.php => @extends('layouts.appadmin')
- 在layouts底下新增appadmin.blade.php
- 在appadmin.blade.php write code

## **CRUD Operations for Admin Panel Backend Services(Posts)**

### Intro to CRUD Backedn Services

- Display all posts data from database
- Create New Post with validation and images upload
- Using tinyMCE to post and post update
- Paginations
- Single paga on frontend page
- Comment system
- Update Posts
- Delete Posts
- Simple Search Method

At the previews lessons we have create Bootstrap admin panel  
Video's link available on description

1. index.blade.php, dashboard.blade.php 修改code

2. create new folder (posts) in admin directory

we will stored all "posts" service files here

allposts.blade.php for stored all post data

create.blade.php for "create new post" page

edit.blade.php for "update post" page

3. we will create new controller (PostsController) and stored in Auth directory

>> `php artisan make:controller Auth\PostsController --resource`

執行時不會自動跑到auth目錄底下，所以手動更改檔名、移到auth目錄裡和namespace

we will store all post services in this controller

4. All Posts

- wirte code --- `Auth\PostsController index()`
- create new routes
- create view for all posts page, write code (allposts.blade.php)

<http://localhost:8000/admin/posts/allposts>

- we must install form and html into project  
<http://www.hc-kr.com/2016/09/laravel-5-tutorial-forms-html-class.html>  
`"laravelcollective/html": "5.3.*" --- composer.json`  
`>> composer update`



6. We will create Edit pages

- copy create.blade.php code to edit.blade.php and write code
- add new routes
- we will create new function for edit and update post data

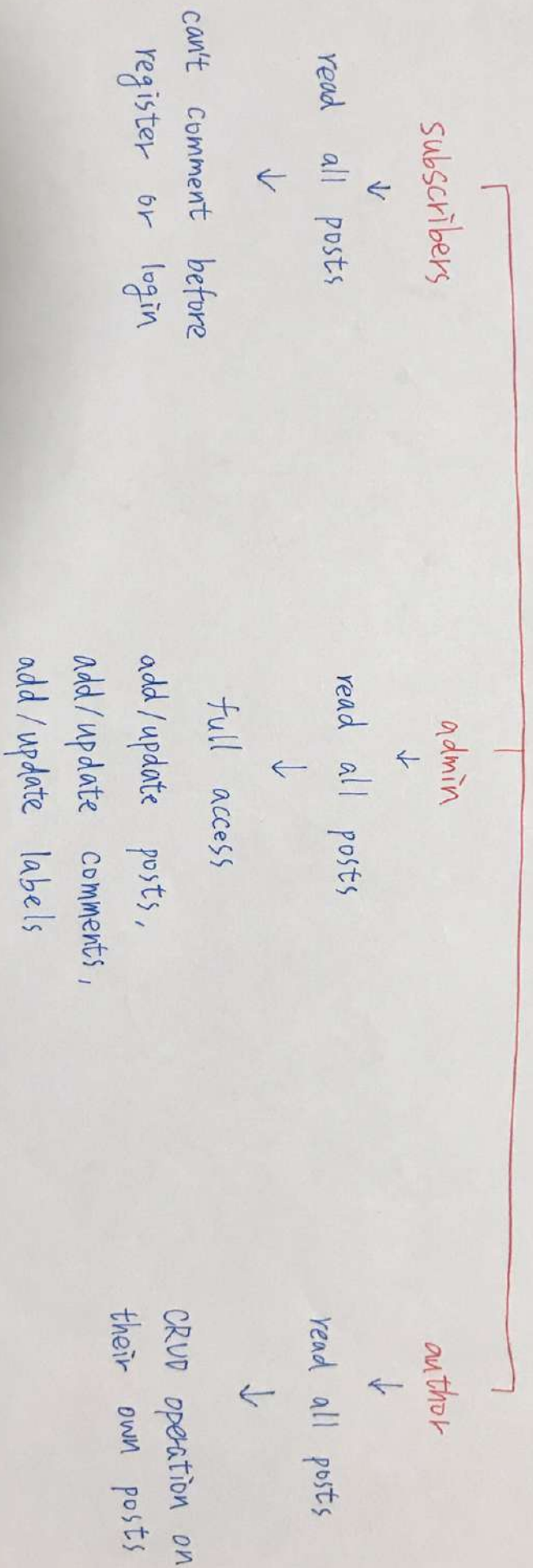
<http://localhost:8000/admin/index>

7. we will create delete post function

- add new routes
- write code --- Auth/PostsController.php destroy()

<http://localhost:8000/admin/posts/allposts>

# Build Blog System using Laravel 5.3





\* Route ch5, 單元5, ch3

\* Controller ch6, 單元9, ch3

\* View & Blade ch7, 單元10, ch4

\* Model { Migration (資料表和欄位) ch8-1, 單元7, ch8  
Eloquent (ORM) (tinker) CRUD資料與關聯 ch8-2  
Seed & Factory 產生測試資料 ch8-3, ch8

\* { Form 傳統的 post / GET ch9-1, 單元11, ch3  
Request ch6 HTTP的請求、回應, 取得資料 ch9-4  
Validate 資料驗證 ch9-2, 9-3, 單元9, ch10, ch6

\* { Middleware ch10-1, ch10

\* { auth ch10-2, 單元11, ch9

\* cookie, session

ch9-4, 單元11, 16, ch14

\* log ch11, 單元9

\* 第三方套件 ch12-3, 單元18

\* 專題實作 (Ajax, jQuery, Bootstrap)