# 上傳檔案前置作業

    {!! Form::open(['method'=>'POST', 'action'=>'App\Http\Controllers\PostsController@store', 'files'=>true])!!}



新增’files’=>true，laravelcollective/html就會自動產生enctype=”multipart/form-data”，也就可以上傳檔案

<form action="/posts" method="POST" enctype="multipart/form-data">

原先的form應該要設置成這樣

        <div class="form-group">

            {!! Form::file('file', ['class'=>'form-control']) !!}

        </div>

新增可以上傳的功能

Form::file()上傳檔案，第一個變數是input\_file的name，第二個變數是class元素

當檔案上傳，可以到posts.store進行設定

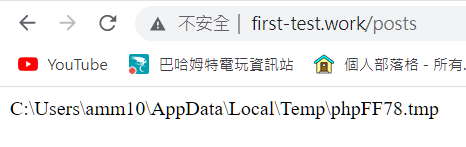
    public function store(CreatePostRequest $request)

    {

        return $request->file('file');

    }

透過$request->file()獲取檔案資訊



會回傳檔案路徑並且產生隨機字串取代檔案名稱

# 解析$request->file內容

    public function store(CreatePostRequest $request)

    {

        $file = $request->file('file');

        echo $file;

        echo "<br>";

        echo $file->getClientOriginalName();

        echo "<br>";

        echo round($file->getSize()/1024, 2)."KB";

        echo "<br>";

        echo round($file->getSize()/1024/1024, 2)."MB";

    }

獲取使用者上傳的file檔案

回傳檔案路徑+隨機名稱

回傳檔案原始名稱

回傳檔案大小/1024 -> Kb

回傳檔案大小/1024/1024 -> Mb

# 同步到資料庫

    public function store(CreatePostRequest $request)

    {

        $input = $request->all();

        if($file = $request->file('file')){

            $name = $file->getClientOriginalName();

            $file->move('images', $name);

            $input['path'] = $name;

        }

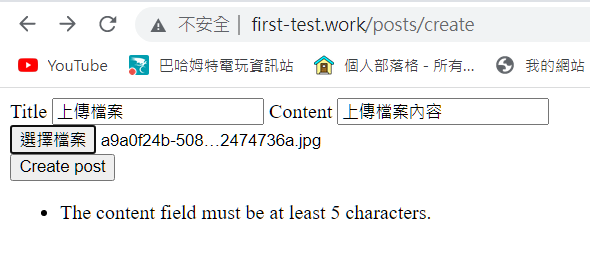
        Post::create($input);

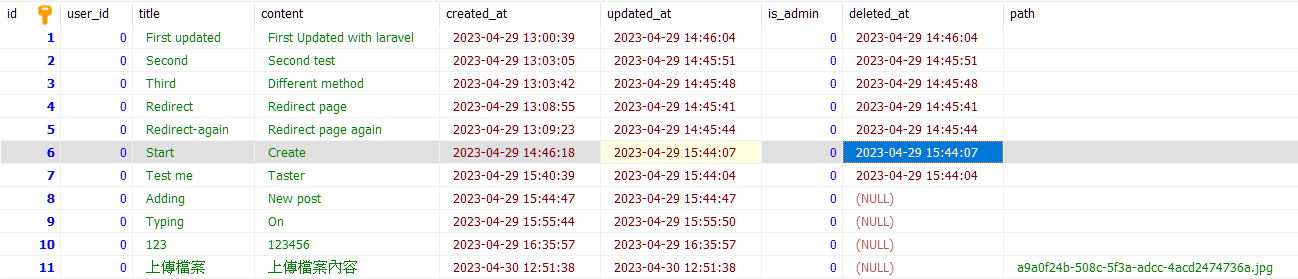
    }

先使用$request->all()獲取所有使用者上傳的資訊 => title, content, file

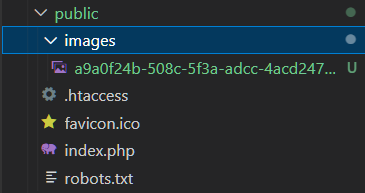
將$request->file的名稱修改成原始檔案的名稱，並且覆蓋$input的$file欄位

最後將修改好的資訊，上傳到database中





有成功上傳到資料庫中



先前有定義$file->move(‘images’, $name); 代表將檔案複製到images的資料夾中，並且名稱為$name

若沒有images資料夾，會自動產生一個新的資料夾

# 存取子 -> 設置圖片路徑

透過前一章節學習到的Accessors，可以將資料庫獲取到的資料進行前處理

    public $directory = '/images/';

    public function getPathAttribute($value)

    {

        return $this->directory.$value;

    }

透過get => 獲取 Path => 哪個欄位 Attribute => 的屬性，將該屬性進行前處理

回傳當前物件的$directory => 設置為先前創立的images資料夾，串接$value的資料

這樣的路徑就變為dynamic的路徑，而不是static

            <div class="image-container">

                <img src="{{$post->path}}" alt="">

            </div>

設置完畢後，可以到index的blade中設置獲取圖片的功能

<img src="{{$post->path}}" alt="">

原先image的src欄位要設置成’/images/{{$post->path}}’，因為有設置accessors的原因，資料庫獲取到的path路徑已經先被前處理，因此獲取到的路徑就與’/images/{{$post->path}}’依樣。