Trong bài này, chúng ta sẽ tiếp tục khám phá tính năng quản lý package trong framework Laravel. Trong phạm vi bài viết, chúng ta sẽ xem xét tỉ mỉ một ví dụ thực tế để minh chứng cho mục đích của bài viết.

Quản lý package trong Laravel là một tính năng quan trọng cho phép bạn đóng gói một nhóm chức năng để có thể phân phối dễ dàng. Hơn thế nữa, bạn có thể luôn luôn xuất bản package của bạn lên những repo như Packagist và Github, cho phép những người phát triển khác sử dụng.

Để chứng minh cho khái niệm này, chúng ta sẽ xây dựng một trang trong Laravel để tải một hình ảnh lên Amazon S3 Cloud. Thay vì như cách thông thường, chúng ta sẽ phát triển như là một package được đóng gói, có thể phân phối và bảo trì dễ dàng.

Trước khi tiếp tục, tôi giả định rằng bạn quen thuộc với Laravel rồi nên tôi sẽ không đi sâu chi tiết vào các khái nhiệm cơ bản của Laravel.

Bạn cũng cần phải có một tài khoản AWS và thông tin để truy cập vào Amazon API để làm việc với ví dụ trong bài viết. Vì thể, hãy chắc rằng bạn đã thiết lập nó trước tiên.

Với mọi thứ đã có đủ, chúng ta sẵn sàng dấn thân vào phần phát triển thực sự.

**Thiết lập các file cho Package**

Hãy xem qua thật nhanh các file chúng ta sẽ triển khai xuyên suốt quá trình của bài viết.

* composer.json: chúng ta cần phải thêm class mapping của package trong tập tin có sẵn composer.json trong thư mục gốc của package.
* config/app.php: đây là một file có sẵn, chúng ta sẽ dùng để thêm vào các phần dịch vụ để có thể tải các views và module sử dụng file này.
* composer.json: Đây là file composer.json chuyên về package mà bạn muốn phân phối package cho các người khác.
* packages/envato/aws/src/Providers/AwsServiceProvider.php: File cung cấp các package của Laravel, được dùng để tải những đối tượng khác của package.
* packages/envato/aws/src/routes/web.php: nó nạp những route mới cho package của chúng ta.
* packages/envato/aws/src/Controllers/AwsController.php: Đây là file contronller (điều hướng) để xử lý logic của ứng dụng cho package của chúng ta.
* packages/envato/aws/src/views/upload.blade.php: File view để xử lý phần logic hiển thị trang.

Đừng lo quá nếu vẫn chưa rõ về nó, chúng ta sẽ thảo luận mọi thứ thật chi tiết trong xuyên suốt quá trình.

**Thiết lập các yêu cầu tiên quyết**

Như chúng ta đã thảo luận trước đó, package của chúng ta sẽ triển khai tình huống tải file lên Amazon S3 Cloud. Trong phần này. chúng ta cần tỉ mỉ xem xét các điều kiện tiên quyết để vận hành package thành công.

Là một người phát triển Laravel, bạn phải quen thuộc với Flysystem, nó cung cấp một abstraction layer tốt để tương tác với filesystem. Nó cung cấp những driver dễ dàng sử dụng để bạn có thể tương tác dễ dàng với bất kỳ loại filesytem nào mà bạn đang phải đối mặt - dù với các file hệ thống nội bộ hoặc hệ thống của AWS S3 Cloud.

Để kích hoạt hỗ trợ của hệ thống file của Amazon S3 Cloud với Flysystem, bạn cần phải cài đặt adapter tương ứng trong package của composer.

Tiếp tục và chạy lệnh composer bên dưới đây từ thự mục gốc của dự án để cài đặt package flysytem-aws-s3-v3.

|  |  |
| --- | --- |
| 1 | $composer require league/flysystem-aws-s3-v3 |

Khi xử lý thành công lệnh này, giờ bạn có thể sử dụng Flysystem của Laravel để tương tác với file hệ thống của Amazon S3 Cloud theo cùng cách bạn sử dụng với file hệ thống nội bộ.

Giờ hãy nhanh chóng xem config/filesystems.php để xem các cấu hình được cung cấp cho file hệ thông của Amazon S3.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | ...  ...  'disks' => [      'local' => [          'driver' => 'local',          'root' => storage\_path('app'),      ],        'public' => [          'driver' => 'local',          'root' => storage\_path('app/public'),          'url' => env('APP\_URL').'/storage',          'visibility' => 'public',      ],        's3' => [          'driver' => 's3',          'key' => env('AWS\_KEY'),          'secret' => env('AWS\_SECRET'),          'region' => env('AWS\_REGION'),          'bucket' => env('AWS\_BUCKET'),      ],  ],  ...  ... |

Như bạn có thế thấy, cấu hình đã đâu vào đấy cho Amazon S3, giờ chỉ cần phải xét các biến ENV trong file .env.

Tiếp tục và thêm các biến sau đây vào file .env

|  |  |
| --- | --- |
| 1  2  3  4  5 | AWS\_KEY={AWS\_KEY\_VALUE}  AWS\_SECRET={AWS\_SECRET\_VALUE}  AWS\_REGION={AWS\_REGION\_VALUE}  AWS\_BUCKET={AWS\_BUCKET\_VALUE}  AWS\_CDN\_URL={AWS\_CDN\_URL\_VALUE} |

Bạn tất nhiên cần phải thay thế những phần có sẵn bằng các giá trị thật. Giờ bạn sẵn sàng sử dụng adapter của Flysystem AWS S3 cho ứng dụng Laravel của bạn.

**Xem xét các file trong package**

Để tạo một Laravel package riêng cho mình, điều đầu tiên là tạo một cấu trúc thu mục hợp lý tuân theo các quy tắc của hệ thống Laravel. Tôi giả định rằng bạn đã chạy một ứng dụng Laravel cơ bản; thực ra thì ứng dụng *blog* mặc định đã đủ tốt rồi.

Tiếp tục tạo thư mục các packages trong thư mục gốc của ứng dụng. Cân nhắc rằng bạn đang chuẩn bị phân phối package của bạn với những người khác, cấu trúc cần có của package nên là {vendor\_name}/{package\_name}

Tuân theo quy tắc đó, hãy tiếp tục tạo thư mục envato/aws bên dược thư mục packages. Bạn cũng có thể đoán ra, envato là tên của vendor, và aws là tên của package. Cuối cùng, hãy tạo một thư mục packages/envatp/aws/src để chứa các file mã nguồn của package.

Giờ chúng ta cần thông báo cho Laravel về package của chúng ta. Tiếp tục mở file composer.json trong thư mục gốc của ứng dụng Laravel và thêm vào "Envato\\Aws\\": "packages/envato/aws/src" trong mục autoload như hiển thị dưới đây:

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13 | ...  ...  "autoload": {      "classmap": [          "database"      ],      "psr-4": {          "App\\": "app/",          "Envato\\Aws\\": "packages/envato/aws/src"      }  },  ...  ... |

Bạn cùng thấy đấy, namespace Envato\Aws\ tương ứng với thư mục packages/envata/aws/src. Giờ chúng ta chỉ cần chạy lệnh dump-autoload để tái lập lại các mappings tương ứng cho composer.

|  |  |
| --- | --- |
| 1 | $composer dump-autoload |

Giờ bạn có thể sử dụng namespace Envato\Aws\ trong ứng dụng của bạn và nó sẽ lấy đúng các file đúng theo thứ tự.

**File Composer của Package**

Giờ hãy tiếp tục bổ sung một file package đặc trung composer.json để bạn có thể phân phối package của bạn cho packagist.

Đến thư mục packages/envato/aws và chạy lệnh sau đây để tạo file composer.json cho package của bạn.

|  |  |
| --- | --- |
| 1 | $composer init |

Bạn sẽ được thông báo với những câu hỏi thường gặp, do vậy cứ lướt qua và nó sẽ tạo một file composer.json.

Sau tất cả, nó sẽ trông giống như thế này đây.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | {      "name": "envato/aws",      "description": "Example of File Upload to AWS S3 Cloud",      "minimum-stability": "dev",      "require": {}  } |

**Route**

Trong package của chúng ta, chúng ta sẽ tạo một trang đơn giản hiển thị trang thái của file được tải lên. Vì thế chúng ta cần tạo một route tương tác với trang đó.

Hãy tạo một file route tại packages/envato/aws/src/routes/web.php

|  |  |
| --- | --- |
| 1  2 | <?php  Route::get('aws/s3/upload', 'Envato\Aws\Controllers\AwsController@upload'); |

Có cần thêm giải thích nào không nhỉ? Bước kế tiếp dĩ nhiên là tạo một file điều hướng tương tác.

**Điều hướng (controller)**

Hãy tạo một file điều hướng Controller tại packages/envato/aws/src/Controllers/AwsController.php với nội dung sau đây.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  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 | <?php  namespace Envato\Aws\Controllers;    use App\Http\Controllers\Controller;    class AwsController extends Controller  {    public function upload(\Illuminate\Contracts\Filesystem\Factory $storage)    {      // load s3 storage      $awsS3Storage = $storage->disk('s3');        // load local storage      $localStorage = $storage->disk('local');        // default path of local storage "storage/app"      $sourceFileContents = $localStorage->get('test.jpg');        // destination filepath in S3 cloud      $destFilePath = 'test\_new.jpg';        // init vars      $imageUrl = '';      $errorMsg = '';        // upload file to AWS S3      if ($awsS3Storage->put($destFilePath, $sourceFileContents, 'public'))      {        $imageUrl = env('AWS\_CDN\_URL') . env('AWS\_BUCKET') . '/' . $destFilePath;      }      else      {        $errorMsg = 'Oops! Something went wrong :(';      }        // call view      return view('aws::upload', ['imageUrl' => $imageUrl, 'errorMsg' => $errorMsg]);    }  } |

Hãy xem xét thật kỹ file này để hiểu ý nghĩa của từng dòng code.

Chúng ta bắt đầu mọi thứ bằng cách thiết lập một namespace của controller để namespace Envato\Aws\Controllers. Nhớ lại rằng chúng ta đã bổ sung mapping của Envato\Aws vào packages/envato/aws/src trong file composer.json ở thư mục gốc để bạn có thể tìm thấy các files của package.

Kế đến, chúng ta sẽ định nghĩa phương thức upload để thực hiện yêu cầu đồng các files nội bộ với Amazon S3 Cloud. Điều quan trọng nhất cần lưu ý là tham số đầu tiên của phương thức upload yêu cầu phần phụ thuộc \Illuminate\Contracts\Filesystem\Factory. Suốt quá trình xử lý, các phần giao kết phù hợp của Laravel sẽ được bổ sung vào.

Bây giờ chúng ta có thể dùng giá trị của filesystem để tạo các giá trị ổ đĩa theo nhu cầu. Giá trị ổ đĩa trong Laravel là một driver cho phép bạn dễ dàng truy cập vào các file hệ thống cơ bản như là ổ đĩa nội bộ, Amazon S3 Cloud, và những thứ tương tự.

|  |  |
| --- | --- |
| 1  2  3  4  5 | // load s3 storage  $awsS3Storage = $storage->disk('s3');    // load local storage  $localStorage = $storage->disk('local'); |

Để đơn giản, chúng tôi sẽ chuyển các file tĩnh đã có sẵn dưới phần lưu trữ nội bộ mặc định của Laravel, và đường dẫn là storage/app/test.jpg

Bước đầu tiên, hãy lấy nội dung của file nguồn.

|  |  |
| --- | --- |
| 1  2 | // default path of local storage "storage/app"  $sourceFileContents = $localStorage->get('test.jpg'); |

Với mọi thứ đã thiết lập như đã đề cập, bạn có thể đồng bộ một file với Amazon S3 sử dụng phương thức put.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | // upload file to AWS S3  if ($awsS3Storage->put($destFilePath, $sourceFileContents, 'public'))  {    $imageUrl = env('AWS\_CDN\_URL') . env('AWS\_BUCKET') . '/' . $destFilePath;  }  else  {    $errorMsg = 'Oops! Something went wrong :(';  } |

Chắc rằng bạn đã lập đúng các biến môi trường AWS, trong trường hợp mọi thứ không được như mong đợi.

Và việc sau cùng là gọi file view để hiển thị hình ảnh đã đồng bộ và một thông điệp phù hợp.

|  |  |
| --- | --- |
| 1  2 | // call view  return view('aws::upload', ['imageUrl' => $imageUrl, 'errorMsg' => $errorMsg]); |

Dĩ nhiên, chúng ta vẫn chưa tạo ra một file view, và đó chính xác là tất cả về phần kế tiếp.

**View**

Hãy tạo một file view ở packages/envato/aws/src/views/upload.blade.php với nội dung sau đây.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  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  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95 | <!DOCTYPE html>  <html lang="{{ config('app.locale') }}">      <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>Laravel</title>            <!-- Fonts -->          <link href="<https://fonts.googleapis.com/css?family=Raleway:100>,600" rel="stylesheet" type="text/css">            <!-- Styles -->          <style>              html, body {                  background-color: #fff;                  color: #636b6f;                  font-family: 'Raleway', sans-serif;                  font-weight: 100;                  height: 100vh;                  margin: 0;              }                .full-height {                  height: 100vh;              }                .flex-center {                  align-items: center;                  display: flex;                  justify-content: center;              }                .position-ref {                  position: relative;              }                .top-right {                  position: absolute;                  right: 10px;                  top: 18px;              }                .content {                  text-align: center;              }                .title {                  font-size: 84px;              }                .links > a {                  color: #636b6f;                  padding: 0 25px;                  font-size: 12px;                  font-weight: 600;                  letter-spacing: .1rem;                  text-decoration: none;                  text-transform: uppercase;              }                .m-b-md {                  margin-bottom: 30px;              }          </style>      </head>      <body>          <div class="flex-center position-ref full-height">              @if (Route::has('login'))                  <div class="top-right links">                      @if (Auth::check())                          <a href="{{ url('/home') }}">Home</a>                      @else                          <a href="{{ url('/login') }}">Login</a>                          <a href="{{ url('/register') }}">Register</a>                      @endif                  </div>              @endif                <div class="content">                  <div class="title m-b-md">                      File upload to S3 Cloud                  </div>                    <div>                    @if ($imageUrl)                        <img src="{{ $imageUrl }}" width="100"/>                    @else                        <span class="error">{{ $errorMsg }}</span>                    @endif                  </div>              </div>          </div>      </body>  </html> |

Đây là một file view chuẩn mực, hiển thị hình ảnh đã tải lên thành công, hoặc giả là một thông điệp báo lỗi phù hợp.

**Service Provider (cung cấp các dịch vụ)**

Chúng ta gần như đã xong việc với package này khi đã tạo ra các files cần thiết. Bước kế tiếp là tạo ra một dich vụ cung cấp (service provider) vì thế bạn có thể đăng ký route và view cho package của chúng ta.

Hãy tạo ra một service provider tại packages/envato/aws/src/Providers/AwsServiceProvider.php với các nội dung sau:

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  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 | <?php    namespace Envato\Aws\Providers;    use Illuminate\Support\ServiceProvider;    class AwsServiceProvider extends ServiceProvider  {    /\*\*     \* Bootstrap the application services.     \*     \* @return void     \*/    public function boot()    {        // load routes        $this->loadRoutesFrom(\_\_DIR\_\_.'/../routes/web.php');          // load view files        $this->loadViewsFrom(\_\_DIR\_\_.'/../views', 'aws');          // publish files        $this->publishes([        \_\_DIR\_\_.'/../views' => resource\_path('views/vendor/aws'),      ]);    }      /\*\*     \* Register the application services.     \*     \* @return void     \*/    public function register()    {    }  } |

Tất nhiên, bạn có thể tạo ra file service provider bằng cách sử dùng lệnh artisan. Nhưng điều này sẽ yêu cầu thực hiện thêm bước nữa là đưa file từ thư mục app/Providers vào package của chúng ta.

Dù sao, hãy xem xét file Service Provider vừa mới được tạo ra.

Trước tiên, chúng ta nạp các route và view có tương tác với package của chúng ta.

|  |  |
| --- | --- |
| 1  2  3  4  5 | // load routes  $this->loadRoutesFrom(\_\_DIR\_\_.'/../routes/web.php');    // load view files  $this->loadViewsFrom(\_\_DIR\_\_.'/../views', 'aws'); |

Tiếp theo, chúng ta cung cấp hỗ trợ của việc xuất bản các views của package vì thế các người phát triển nếu muốn override các views đó có thể thực hiện chuyện đó. Tiếp theo họ chạy lệnh php artisan vendor:publish, Laravel sao chép các views từ packages/envato/aws/src/views/đến resources/views/vendor/aws.

Bây giờ, họ có thể thay đổi các views trong thư mục resources/views/vendor/aws, và nó sẽ được tự chọn bở Laravel thay vì các views bên trong thư mục packages/envato/aws/src/views/. Thực tế, đây là một cách phù hợp để bổ sung các views của package của bên thứ 3, thay vì trực tiếp thay đổi các views của package.

Như bạn mong đợi. bạn cần phải bổ sung service provider trong config/app.php. Thêm vào phần sau đây trong mảng providers.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13 | ...  ...  /\*   \* Application Service Providers...   \*/  App\Providers\AppServiceProvider::class,  App\Providers\AuthServiceProvider::class,  App\Providers\BroadcastServiceProvider::class,  App\Providers\EventServiceProvider::class,  App\Providers\RouteServiceProvider::class,  Envato\Aws\Providers\AwsServiceProvider::class, // Our package service provider  ...  ... |

Bây giờ mọi thứ đã đâu vào đó hết rồi, vì thế bạn có thể tiếp tục thử nghiệm package của chúng ta.

Tiếp tục chạy đường dẫn URL <http://your-laravel-application/aws/s3/upload> trên trình duyệt của bạn. Nếu mọi thứ diễn ra tốt đẹp, bạn nên thấy hình ảnh trên trang của bạn được nạp từ Amazon S3 Cloud. Hãy cho tôi biết nếu bạn đang vướng phải bất kỳ trở ngại gì, và tôi sẽ vui lòng trả lời chúng.

Vậy chúng ta đang chuẩn bị kết thúc bài viết này, và tôi hy vọng bạn sẽ thích nó