FPT POLYTECHNIC

Lab 8: Mail, Errors & Logging

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hiểu được cấu hinh mail
- ✓ Các cơ chế gởi mail
- ✓ Cấu hình Error & Logging
- ✓ Tạo trang thông báo lỗi

BÀI 1 (3 ĐIỂM)

Để thực hiện việc gởi mail trong Laravel, ta tạo class kề thừa từ “mailable”, cấu hình file “.env”

HƯỚNG DẪN:

- ✓ Thực hiện cấu hình Email trong file .env gồm các thông số MAIL_DRIVER, MAIL_HOST, MAIL_PORT, MAIL_USERNAME, MAIL_PASSWORD, MAIL_ENCRYPTION

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME= // some username
MAIL_PASSWORD= // some password
MAIL_ENCRYPTION=null
```

Đoạn code trên sử dụng mailbox là mailtrap (MAIL_HOST=smtp.mailtrap.io). Trong trường hợp dùng gmail thì phải thay đổi Mail_Host

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=addYourEmail
MAIL_PASSWORD=AddYourEmailPassword
MAIL_ENCRYPTION=tls
```

- ✓ Tạo class tên SendMailable kế thừa Mailable để thực hiện viêc gởi mail

```
php artisan make:mail SendMailable
```

- ✓ File tạo ra sẽ nằm trong đường dẫn App\Mail\SendMailable.php, sinh viên lưu ý thuộc tính “Name”

```php
<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;
use Illuminate\Contracts\Queue\ShouldQueue;

class SendMailable extends Mailable
{
    use Queueable, SerializesModels;
    public $name;

    /**
     * Create a new message instance.
     *
     * @return void
     */
    public function __construct($name)
    {
        $this->name = $name;
    }

    /**
     * Build the message.
     *
     * @return $this
     */
    public function build()
    {
        return $this->view('emails.name');
    }
}
```

- ✓ Tạo file blade view “name.blade.php”để nhận dữ liệu từ $name

```
<div>
    Hi, This is : {{ $name }}
</div>
```

- ✓ Bên trong route cần chỉ định controller xử lý, sinh viên chỉnh sửa file route.php:

```
Route::get('/send/email', 'HomeController@mail');
```

- ✓ Mở HomeController@mail, chỉnh lại code của hàm mail

```
// HomeController.php

use Illuminate\Support\Facades\Mail;
use App\Mail\SendMailable;

public function mail()
{
   $name = 'Krunal';
   Mail::to('fpoly@gmail.com')->send(new SendMailable($name));

   return 'Email was sent';
}
```

Sinh viên lưu ý thay đổi email tồn tại thực tế phần Mail::to('fpoly@gmail.com') và import

```
use Illuminate\Support\Facades\Mail;
use App\Mail\SendMailable;
```

- ✓ Chạy http://localhost:8000/send/email và kiểm tra kết quả “Email was sent”

BÀI 2 (3 ĐIỂM):

Gởi mail xác nhận khi người dùng đăng ký thành viên thành công trên web Laravel

Hướng dẫn:

Sinh viên đăng ký mail https://mailtrap.io/register/signup để sử dụng mailtrap

Các bước cấu hình giống với câu 1, sinh viên thực hiện nhanh như sau:

```
1
2 MAIL_DRIVER=smtp
3 MAIL_HOST=smtp.mailtrap.io
4 MAIL_PORT=2525
5 MAIL_USERNAME=efaea9a7b1713f
6 MAIL_PASSWORD=a5a09892******
7 MAIL_ENCRYPTION=null
8
```

- ✓ Tạo lớp Welcomemail kế thừa Mailable

```
1
2 php artisan make:mail WelcomeMail
3
```

- ✓ Code bên trong WelcomeMail.php

```php

<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;
use Illuminate\Contracts\Queue\ShouldQueue;

class WelcomeMail extends Mailable
{
    use Queueable, SerializesModels;


    public $user;
    /**
     * Create a new message instance.
     *
     * @return void
     */
    public function __construct($user)
    {
        $this->user = $user;
    }

    /**
     * Build the message.
     *
     * @return $this
     */
    public function build()
    {
        return $this->view('emails.welcome');
    }
}

```

- ✓ Bên trong resources/views tạo thư mục “email” bên trong thư muc email tạo welcome.blade.php, lưu ý biến $user có phạm vi Public được khai báo

trong lớp WelcomeMail.php, biến này được truy cập trực tiếp trong view

```
1
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <title>Welcome Email</title>
6  </head>
7
8  <body>
9  <h2>Welcome to the site {{$user['name']}}</h2>
10 <br/>
11 Your registered email-id is {{$user['email']}}
12 </body>
13
14 </html>
15
```

- ✓ Để gởi mail, tùy chỉnh code trong controller RegisterController.php, cụ thể chỉnh code phương thức Create

```php

<?php

namespace App\Http\Controllers\Auth;

use ...
use .....
use App\Mail\WelcomeMail;
use Illuminate\Support\Facades\Mail;


class RegisterController extends Controller
{
    .
    ...
    ....
```

```php
    protected function create(array $data)
    {
        $user =  User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => bcrypt($data['password']),
        ]);

        Mail::to($data['email'])->send(new WelcomeMail($user));

        return $user;
    }
}

```

- ✓ Sinh viên chạy và tiến hành đăng ký thành viên (sinh viên cần xem các bài lab trước để biết cách bật tính năng đăng ký thành viên trong Laravel https://laravel.com/docs/5.6/authentication)

FPT POLYTECHNIC

BÀI 3 (3 ĐIỂM):

Quản lý Exception trong Laravel

- ✓ Sinh viên mở file config/app.php

```
...
...
/*
|--------------------------------------------------------------------------
| Application Debug Mode
|--------------------------------------------------------------------------
|
| When your application is in debug mode, detailed error messages with
| stack traces will be shown on every error that occurs within your
| application. If disabled, a simple generic error page is shown.
|
*/

'debug' => env('APP_DEBUG', false),
...
...
```

Sinh viên chú ý APP_DEBUG là mức độ quản lý exception được cấu hình bên trong file .env. Trong môi trường người lập trình thường thiết lập thông số này có giá trị "true" giúp debug được khi có lỗi xảy ra

- ✓ Để xem các lỗi xảy ra, laravel cũng cho phép lưu vết lại trong file log, điều chỉnh code bên trong config/app.php

```
...
...
'log' => env('APP_LOG', 'single'),

'log_level' => env('APP_LOG_LEVEL', 'debug'),
...
...
```

Mặc định file log lưu tại storage/logs/laravel.log

- ✓ Laravel cung cấp mặc định lớp Handler trong app/Exceptions/Handler.php

```php
<?php

namespace App\Exceptions;

use Exception;
use Illuminate\Auth\AuthenticationException;
use Illuminate\Foundation\Exceptions\Handler as ExceptionHandler;

class Handler extends ExceptionHandler
{
    /**
     * A list of the exception types that should not be reported.
     *
     * @var array
     */
    protected $dontReport = [
        \Illuminate\Auth\AuthenticationException::class,
        \Illuminate\Auth\Access\AuthorizationException::class,
        \Symfony\Component\HttpKernel\Exception\HttpException::class,
        \Illuminate\Database\Eloquent\ModelNotFoundException::class,
        \Illuminate\Session\TokenMismatchException::class,
        \Illuminate\Validation\ValidationException::class,
    ];

```

```php
    /**
     * Report or log an exception.
     *
     * This is a great spot to send exceptions to Sentry, Bugsnag, etc.
     *
     * @param  \Exception  $exception
     * @return void
     */
    public function report(Exception $exception)
    {
        parent::report($exception);
    }

    /**
     * Render an exception into an HTTP response.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Exception  $exception
     * @return \Illuminate\Http\Response
     */
    public function render($request, Exception $exception)
    {
        return parent::render($request, $exception);
    }
```

```php

    /**
     * Convert an authentication exception into an unauthenticated response.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Illuminate\Auth\AuthenticationException  $exception
     * @return \Illuminate\Http\Response
     */
    protected function unauthenticated($request, AuthenticationException $exception)
    {
        if ($request->expectsJson()) {
            return response()->json(['error' => 'Unauthenticated.'], 401);
        }

        return redirect()->guest(route('login'));
    }
}
```

Trong đoạn code trên sinh viên chú ý hàm "report"

```php
/**
 * Report or log an exception.
 *
 * This is a great spot to send exceptions to Sentry, Bugsnag, etc.
 *
 * @param  \Exception  $exception
 * @return void
 */
public function report(Exception $exception)
{
    parent::report($exception);
}
```

Được dùng log các lỗi ra file log trong khi hàm "render" sẽ xuất lỗi ra màn hình

- ✓ Chúng ta sẽ tạo một Exception theo ý người dùng, không dùng exception mặc định, sinh viên tạo app/Exceptions/CustomException.php

```php
<?php

namespace App\Exceptions;

use Exception;

class CustomException extends Exception
{
    /**
     * Report the exception.
     *
     * @return void
     */
    public function report()
    {
    }

    /**
     * Render the exception into an HTTP response.
     *
     * @param  \Illuminate\Http\Request
     * @return \Illuminate\Http\Response
     */
    public function render($request)
    {
        return response()->view(
                'errors.custom',
                array(
                    'exception' => $this
                )
        );
    }
}
```

Lớp CustomException kế thừa lớp Exception

Khi có error thì user được chuyển tới trang errors.custom

- ✓ Tạo view resources/views/errors/custom.blade.php

```
Exception details: <b>{{ $exception->getMessage() }}</b>
```

- ✓ Thay đổi code trong hàm render của app/Exceptions/Handler.php

```
...
...
/**
 * Render an exception into an HTTP response.
 *
 * @param  \Illuminate\Http\Request  $request
 * @param  \Exception  $exception
 * @return \Illuminate\Http\Response
 */
public function render($request, Exception $exception)
{
    if ($exception instanceof \App\Exceptions\CustomException)  {
        return $exception->render($request);
    }

    return parent::render($request, $exception);
}
...
...
```

- ✓ Tạo controller app/Http/Controllers/ExceptionController.php

```
<?php
namespace App\Http\Controllers;

use App\Http\Controllers\Controller;

class ExceptionController extends Controller
{
    public function index()
    {
        // something went wrong and you want to throw CustomException
        throw new \App\Exceptions\CustomException('Something Went Wrong.');
    }
}
```

- ✓ Cuối cùng là tạo route để trình duyệt chạy

```
// Exception routes
Route::get('exception/index', 'ExceptionController@index');
```

FPT POLYTECHNIC

- ✓ Chạy kiểm tra chương trình http://your-laravel-site.com/exception/index

BÀI 4 (1 ĐIỂM) : Giảng viên cho thêm