Email varification

Since version 5.7 Laravel has introduced Email verification and Account activation out of the box. These features force the registered users to activate their accounts before accessing certain routes in the application.

These feature is not active by default. To make it active you can follow these steps.

Step 1 - Implement `mustVerifyEmail` interface in the User model

Open your `User` model and implement the `mustVerifyEmail` interface like this:

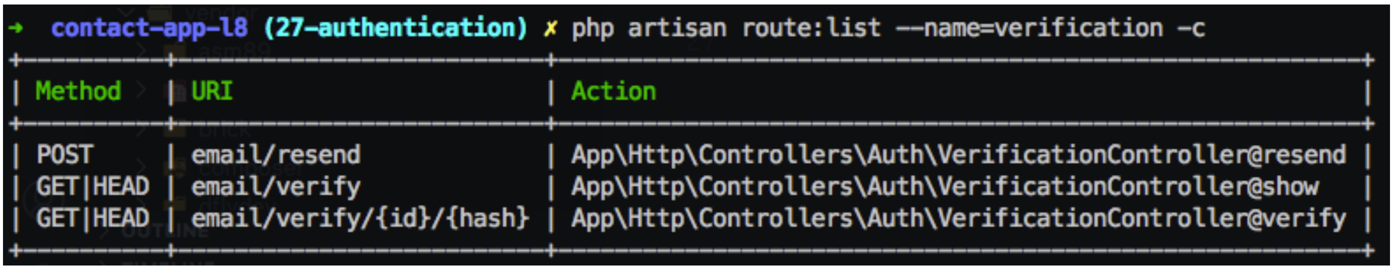
```php
class User extends Authenticatable implements MustVerifyEmail
{
    use Notifiable;
    ...
}
```

Step 2 - Enable Verification Route

Open your `routes/web.php` and specify `verify` option in `Auth::routes` as follow:

```php
Auth::routes(['verify' => true]);
```

Now if you go to your terminal and type `php artisan route:list --name=verification -c` you'll see three routes listed.



```
➜  contact-app-l8 (27-authentication) ✗ php artisan route:list --name=verification -c
+----------+---------------------------+---------------------------------------------------------+
| Method   | URI                       | Action                                                  |
+----------+---------------------------+---------------------------------------------------------+
| POST     | email/resend              | App\Http\Controllers\Auth\VerificationController@resend |
| GET|HEAD | email/verify              | App\Http\Controllers\Auth\VerificationController@show   |
| GET|HEAD | email/verify/{id}/{hash}  | App\Http\Controllers\Auth\VerificationController@verify |
+----------+---------------------------+---------------------------------------------------------+
```

Step 3 - Protect your route

If you open `app/Http/Kernel.php` then go to `routeMiddleware` property, you will find the `verified` middleware which is map to `EnsureEmailIsVerified`.

```php
protected $routeMiddleware = [
  // ...
  'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
];
```

You can protect particular route(s) by using the middleware in `web.php` file like so:

```
Route::middleware(['auth', 'verified'])->group(function () {
    Route::get('/contacts', 'ContactController@index')->name('contacts.index');
    Route::post('/contacts', 'ContactController@store')->name('contacts.store');
    // other contact related routes
    Route::get('/dashboard', 'HomeController@index')->name('home');
});
```

Or you can do that in the controller's constructor like this:

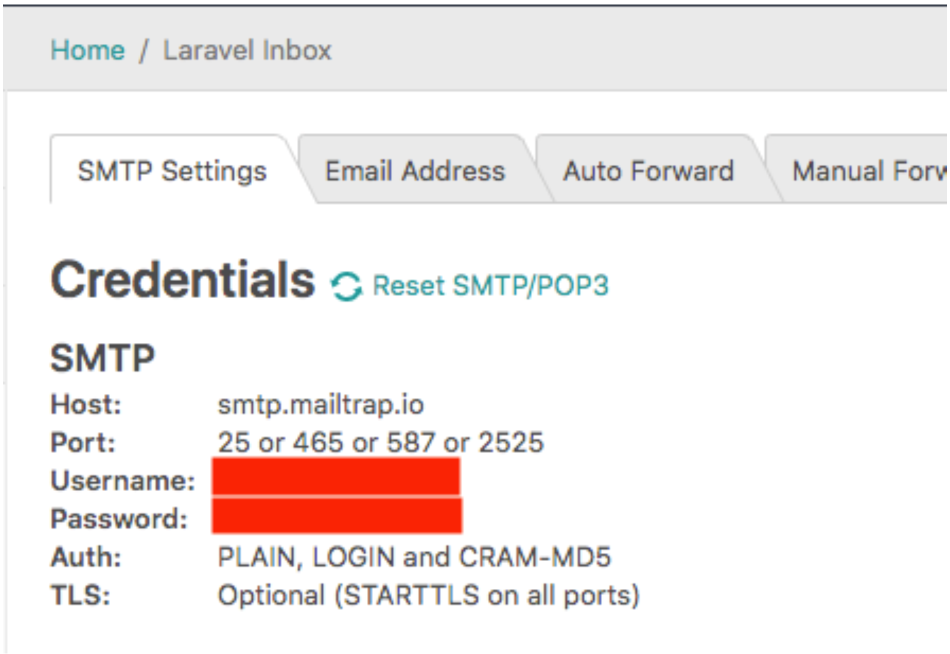
```
// ContactController.php
class ContactController extends Controller
{
    public function __construct()
    {
        $this->middleware(['auth', 'verified']);
    }
    // ...
}

// HomeController.php
class HomeController extends Controller
{
    public function __construct()
    {
        $this->middleware(['auth', 'verified']);
    }
}
```

Step 4 - Setup Email Credentials

In the previous lesson we've set the `MAIL_MAILER` to the `log` to catch the email that is going to be sent inside the `*.log` file. But for more realistic scenario you can set it back to `smtp`. By default Laravel use `mailhog` (formerly `smtp.mailtrap.io`) as the host address of the SMTP server.

In order to make it work you need to install the mailhog in your machine. However in this guide we'll be using mailtrap.

During your development using **mailtrap** to capture your email is perfectly fine. Just go to `mailtrap.io` then register an account. Once you've registered you can choose your inbox, then go to **SMTP Settings**.



Credentials

SMTP

Host: smtp.mailtrap.io
Port: 25 or 465 or 587 or 2525
Username: [redacted]
Password: [redacted]
Auth: PLAIN, LOGIN and CRAM-MD5
TLS: Optional (STARTTLS on all ports)

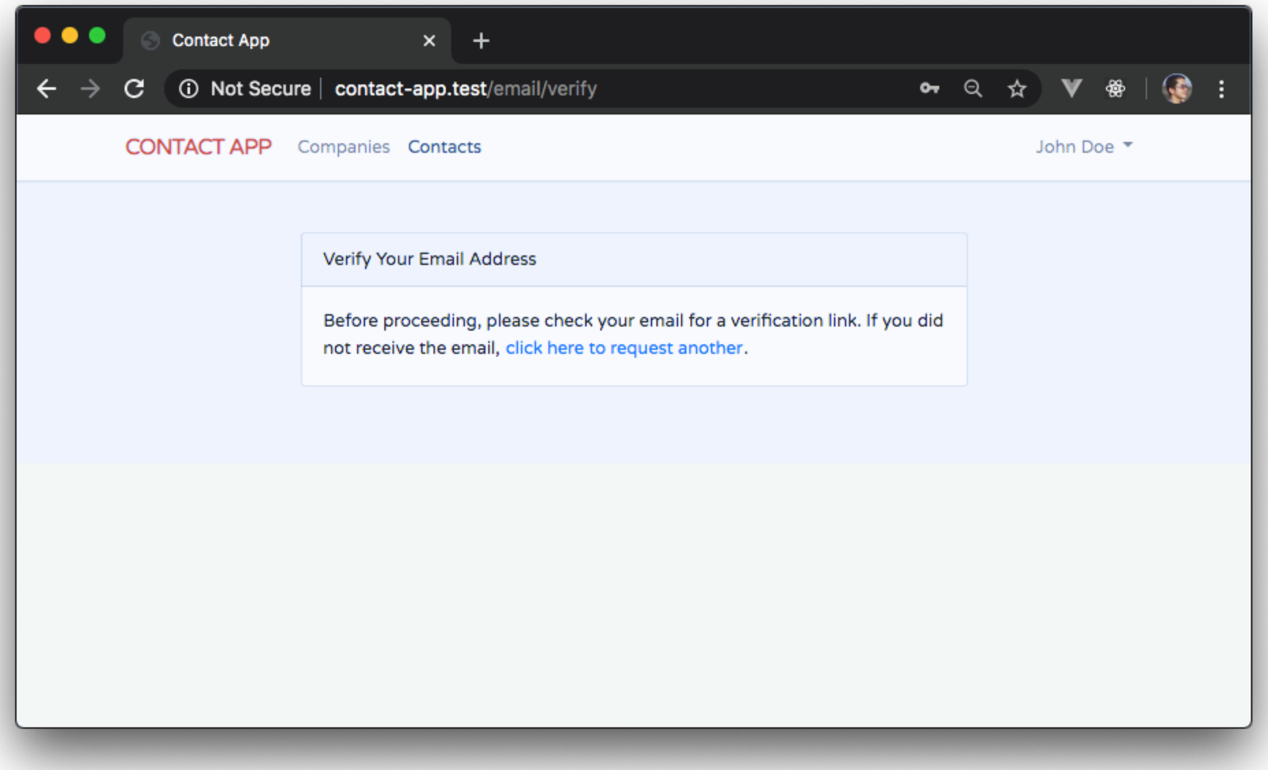
Copy username and password in the **credentials** section and paste to your `.env` file.

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=your-user-email
MAIL_PASSWORD=your-password-email
MAIL_ENCRYPTION=null
```

Save all changes and don't forget to restart your server if you're using php built-in server to apply the changes.

Step 5 - Test Email Verification functionality

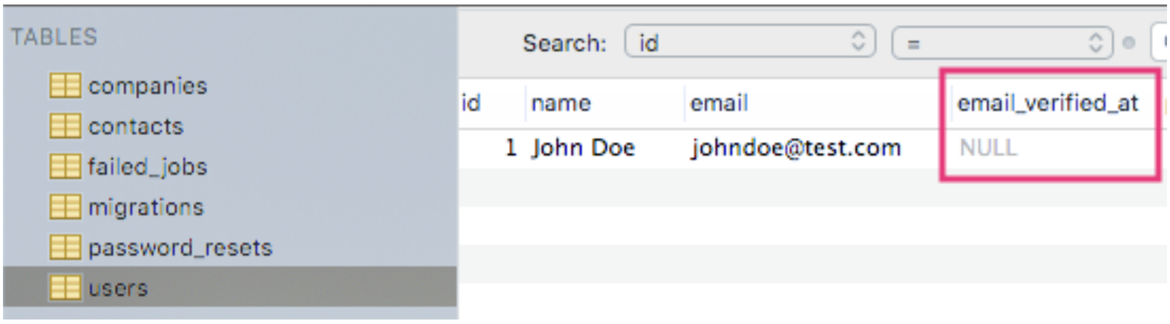
You can test the Email Verification functionality by accessing the **All Contacts** page. Since you applied the `verified` middleware you'll be redirected to **Verify Your Email Address** page.



Verify Your Email Address

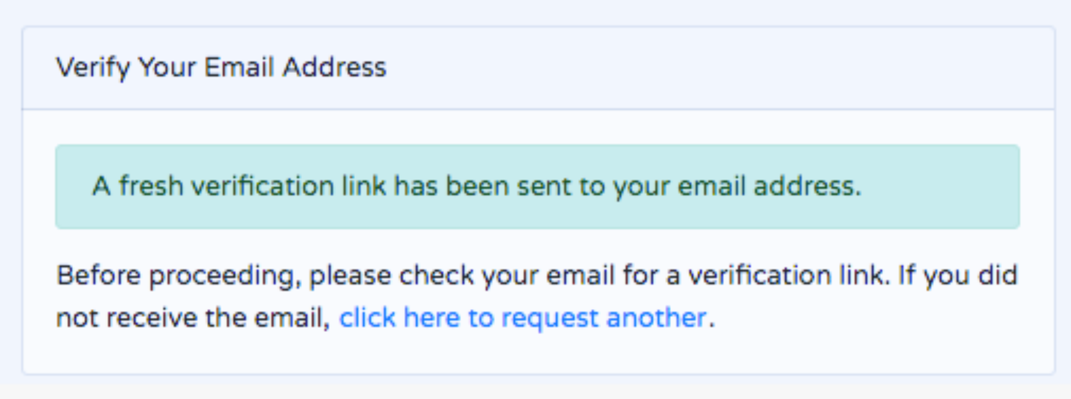
Before proceeding, please check your email for a verification link. If you did not receive the email, click here to request another.

Under the hood Laravel authentication will check the `email_verified_at` of the current user in the `users` table. If it was empty it means you have not verified your email.



id	name	email	email_verified_at
1	John Doe	johndoe@test.com	NULL

You can hit the **click here to request another** link to send a verification link to your email.

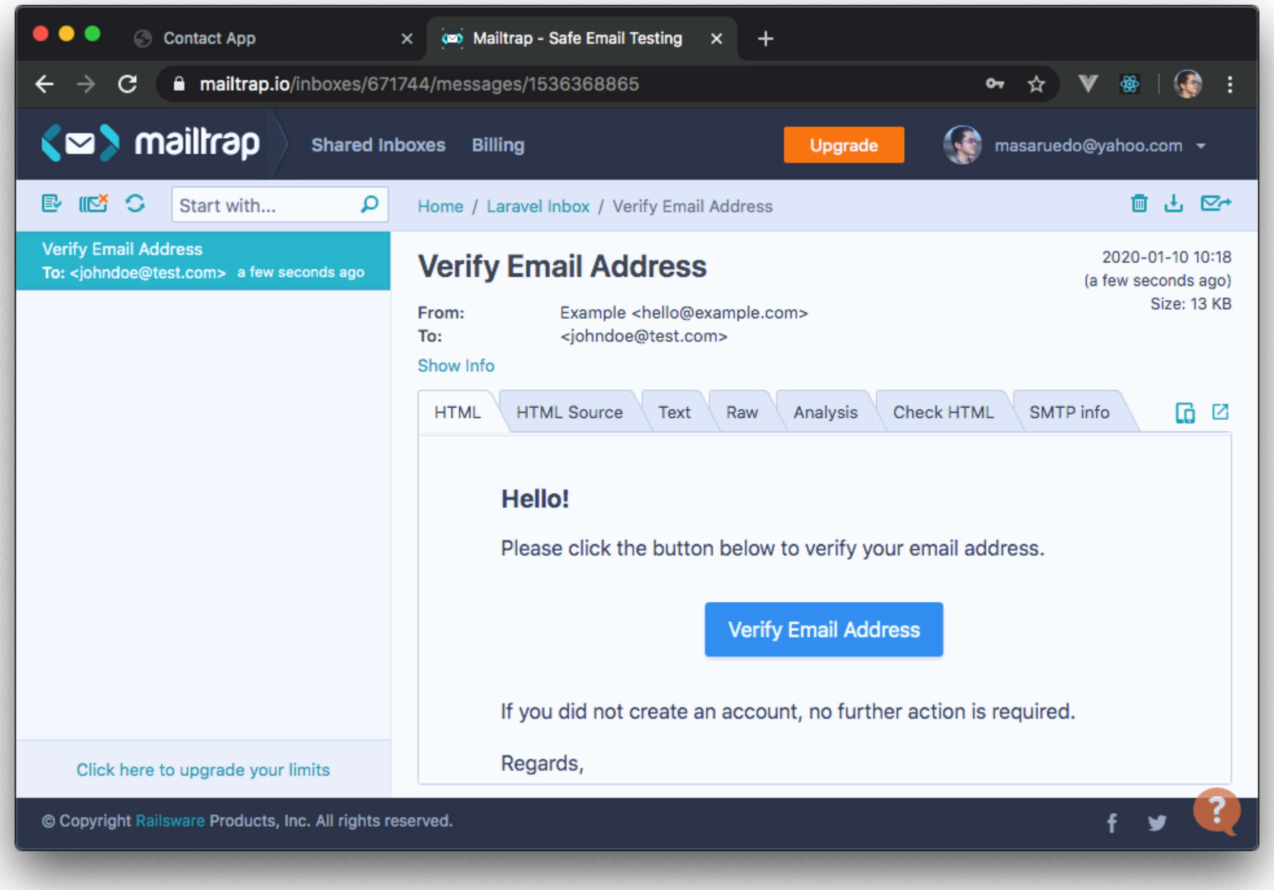


Verify Your Email Address

A fresh verification link has been sent to your email address.

Before proceeding, please check your email for a verification link. If you did not receive the email, click here to request another.

If you open your inbox in mailtrap, you'll find verification email comes in. You can then hit the **Verify Email Address** button to verify your account.


Contact App
Mailtrap - Safe Email Testing
mailtrap.io/inboxes/671744/messages/1536368865
mailtrap
Shared Inboxes
Billing
Upgrade
masaruedo@yahoo.com
Start with...
Home / Laravel Inbox / Verify Email Address
Verify Email Address
To: <johndoe@test.com> a few seconds ago
Show Info
HTML
HTML Source
Text
Raw
Analysis
Check HTML
SMTP info
Verify Email Address
Click here to upgrade your limits
© Copyright Railsware Products, Inc. All rights reserved.

Verify Email Address

2020-01-10 10:18
(a few seconds ago)
Size: 13 KB

From: Example <hello@example.com>
To: <johndoe@test.com>

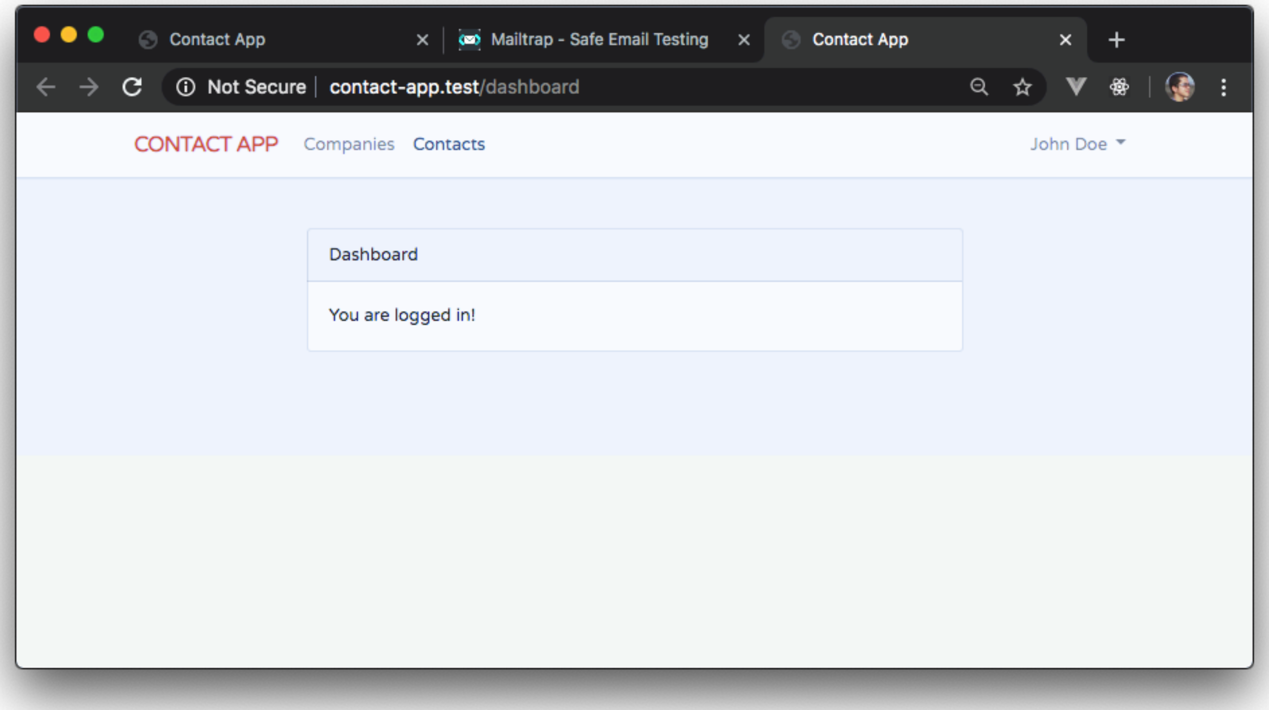
Hello!

Please click the button below to verify your email address.

If you did not create an account, no further action is required.

Regards,

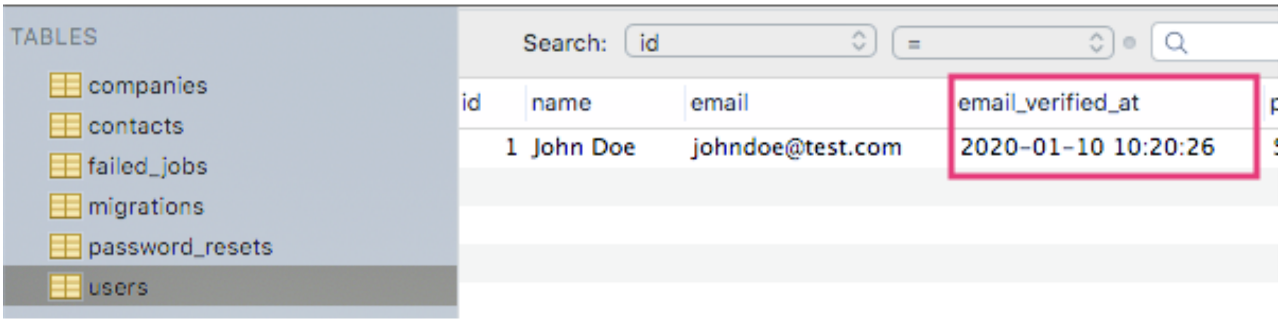
Once you hit that link your account will be verified and you'll be taken to the `dashboard` page.



Dashboard

You are logged in!

Also if you take a look at `users` table, the `email_verified_at` column now has been set with date and time where you verified your email.



TABLES

- companies
- contacts
- failed_jobs
- migrations
- password_resets
- users

Search: id =

id	name	email	email_verified_at
1	John Doe	johndoe@test.com	2020-01-10 10:20:26