

# Mail and Email Verification



Hadeel Adres



# Content

**1**

Introduction

**2**

Send Email

**3**

Live Demo

**4**

Email Verification

**5**

Live Demo

**6**

Sources

# Introduction



- Laravel provide mail class to send email.
- You can use several drivers for sending email in laravel.
- You can use smtp, Mailgun, Postmark, Amazon SES, and sendmail.
- You have to configure on **env** file what driver you want to use.

- Laravel already provide email verification feature by default
- You can do your own logic with manually email verification process
- making account verification or email verification will prevent users from entering or registering fake emails when registering.



**Send Email**



# Step 1: Make Configuration

Configure your mailing configuration by setting your values in the **.env** file in the root directory of your project.

**For Example:**

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=5238b6124f43cb
MAIL_PASSWORD=89472f76621035
MAIL_ENCRYPTION=tls
```

# Step 2: Create Controller & Mail Class

**php artisan make:controller MailTester**

To send a message, we use the **to** method on the **Mail** facade.



```
1 use Illuminate\Support\Facades\Mail;
```



```
1 $details=[
2     'title' => 'Mail from surfside Media',
3     'body' => 'This is for testing mail using gmail'
4 ];
5
6
7 Mail::to("ex@gmail.com")->send(new SendEmail($details));
```



## Different Methods of mailing System:


Functions which helps to perform the mailing interaction better.

- `$message → subject('Laravel Tutorial');`
- `$message → from('testuser@example.com', 'anyname');`
- `$message → to('alex@example.com', 'Alex');`
- `$message → sender('testuser@example.com', 'anyname');`
- `$message → returnPath('testuser@example.com');`
- `$message → cc('testuser@example.com', 'anyname');`

## php artisan make:mail SendEmail

In this step we will create mail class SendEmail for email sending.

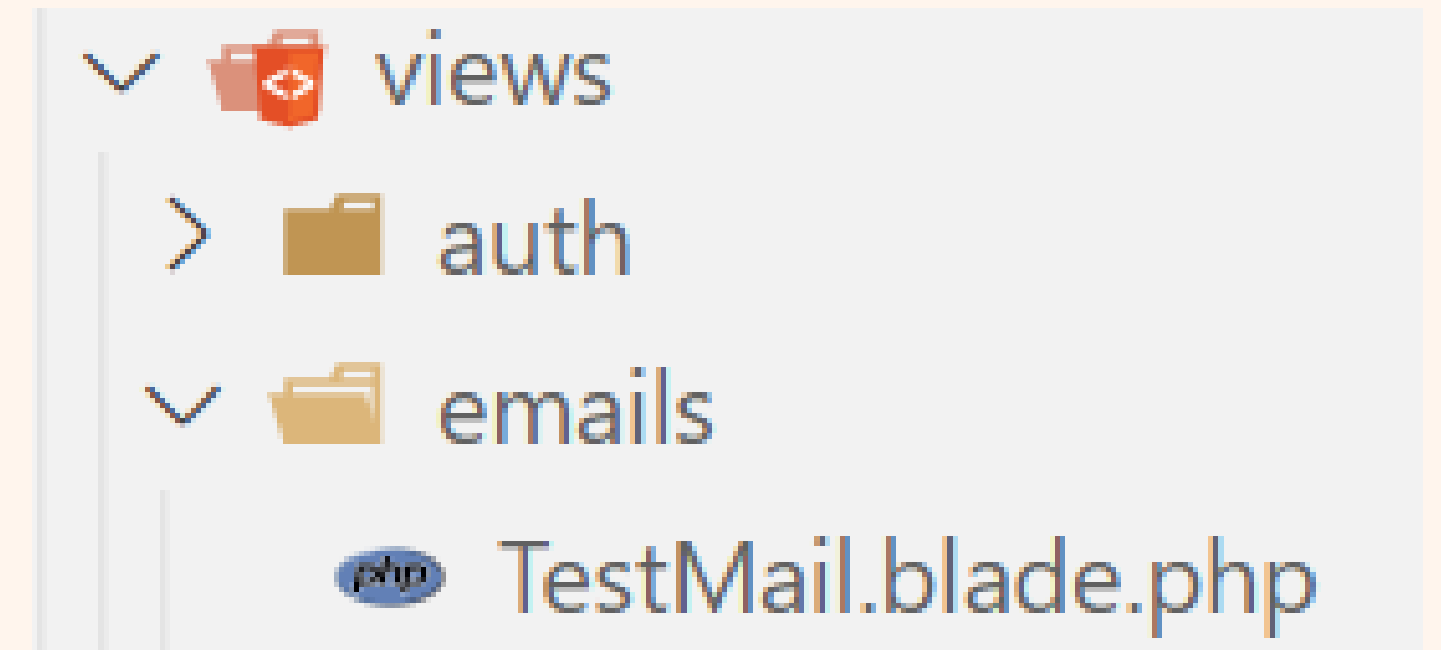
With mailable class' **build** method, you may use the view method to specify which template should be used when rendering the email's contents.



```
1 public function build()  
2     {  
3         return $this->subject('Test Mail From Surfside Media')->view('emails.TestMail');  
4     }
```

## Step 3: Create Blade View

We will create blade view file and write the email that we want to send, and we just write some HTML and CSS Code.



## Step 4: Add Route

We create web route for testing send email

```
Route::get('/send-email', [MailTester::class, 'sendEmail']);
```

**Controller**



**Function in our Controller**



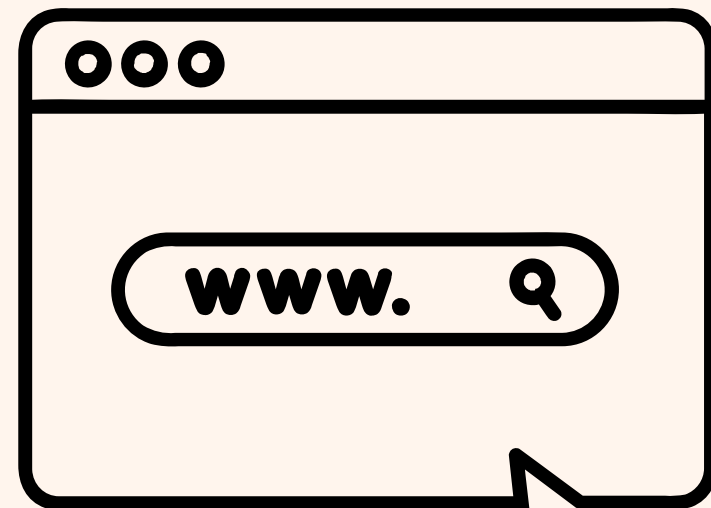
**Live Demo**

# Email Verification





# Email Verification - What happening under the hood



`http://my-website/verify-account?code=qCZ4q3Cmhbq1RHM`



`qCZ4q3Cmhbqlo1RH7MZta`

# Step 1: User Model

We verify that our **App\Models\User** model implements the **Illuminate\Contracts\Auth\MustVerifyEmail** contract.



```
1 use Illuminate\Contracts\Auth\MustVerifyEmail;
```



```
1 class User extends Authenticatable implements MustVerifyEmail
```


## Step 2: Edit the Route



```
1  Auth::routes([  
2      'verify' => true  
3  ]);
```

# Step 3: Add a Middleware

- We need the Middleware to check that user is verified or not
- verified come by default with Laravel in **App\Http\Middleware\Kernel.php**



```
1 -> middleware('verified');
```



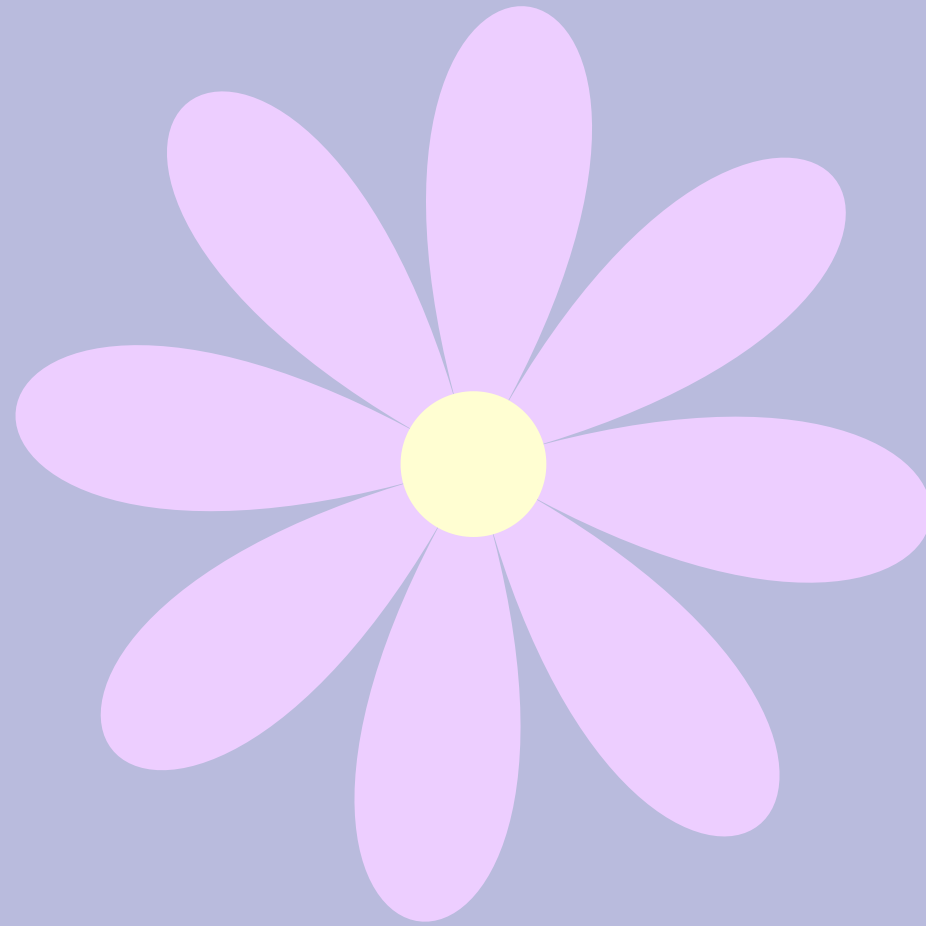
```
1 'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
```

**Live Demo**



# Sources

- <https://laravel.com/docs/9.x/mail#sending-mail>
- <https://www.youtube.com/watch?v=kWEvrHVg8kl>
- <https://www.youtube.com/watch?v=XWTNSEM5ylc>
- <https://laravel.com/docs/9.x/verification#verification-email-customization>



**Thank you for your attention**