

## Jesús Chicano

Blog personal sobre desarrollo web, aplicaciones útiles y muchos tutoriales y manuales.



☰ Menú

# Manual de Laravel en español

Cómo enviar e-mails



Laravel 5.6

Jesús Chicano

## Cómo mandar e-mails desde Laravel

30 julio, 2018 por Jesús

Podríamos decir que el correo electrónico sigue siendo una de las herramientas más utilizadas en Internet. Todo funciona gracias a él, y todo el mundo tiene al menos uno. Son necesarios para recibir notificaciones, correo personal, suscripciones, acceso a aplicaciones... En definitiva, es muy importante que una aplicación cuente con un sistema de envío de e-mails.

Así pues, en este tutorial vamos a configurar una aplicación que pueda mandar e-mails desde Laravel.

## Servidor de correo

Lo primero que tenemos que tener es un servidor de correo, el cual tenga habilitado el protocolo SMTP o POP3.

Esto hará de baliza para el envío de los correos electrónicos a través de Internet, ya que Laravel no tiene su propio servidor de correo.

Para este ejemplo podemos usar [Mailtrap](#). Mailtrap es un servidor de correo bastante utilizado por los desarrolladores para el testeo de sus aplicaciones. Es muy sencillo y su paquete freemium nos deja una bandeja de entrada de hasta 50 correos.

Una vez nos hemos logueado o creado una cuenta tenemos que copiarnos los datos y credenciales del servidor. Estos se encuentran en «Demo Inbox».

Dentro de la configuración se nos deja escoger entre el tipo de aplicación que va a conectarse al servidor. Si escogemos Laravel, vemos que ya nos genera un código, listo para pegar en nuestro fichero de configuración.

Esta información irá en el fichero `.env` de Laravel.

```
1. MAIL_DRIVER=smtp
2. MAIL_HOST=smtp.mailtrap.io
3. MAIL_PORT=2525
4. MAIL_USERNAME="el usuario que te den"
5. MAIL_PASSWORD="la pass que te genere"
6. MAIL_ENCRYPTION=tls
```

**Nota importante:** Muchos servidores de correo, ya implementan la doble autenticación para acceder. Es posible que con estos datos nos de algún error de conexión y no funcione correctamente.

Para solucionar esto, debemos copiar esta misma configuración y grabarla en el archivo `mail.php` dentro del directorio `config`.

```
1. // solo en el caso de servidores con doble autenticación
2. 'driver' => env('MAIL_DRIVER', 'smtp'),
3. 'host' => 'smtp.mailtrap.io',
4. 'port' => 2525,
5. 'from' => [
6.     'address' => 'from@example.com',
7.     'name' => 'Example',
8. ],
9. 'encryption' => env('MAIL_ENCRYPTION', 'tls'),
10. 'username' => 'tu usuario',
11. 'password' => 'tu pass',
12. 'sendmail' => '/usr/sbin/sendmail -bs',
13. 'pretend' => false,
```

## Preparar la ruta

Para este ejemplo se va a desarrollar una ruta que reciba una función anónima donde utilizaremos la clase **Mail** de Laravel.

Mail cuenta con un método llamado *send()* que recibe 3 parámetros, a saber: vista/plantilla del correo electrónico, los datos que queremos enviar, y una función anónima.

La vista podemos prepararla dentro del directorio */views/emails* y allí crear un archivo llamado *notificacion.blade.php*

```
1. // esta es la vista del correo
2. // podemos usar etiquetas HTML
3.
4. Visita mi web <a href="{{ $link }}">{{ $link }}</a> para ver muchos más
   tutoriales.
```

Esta vista en especial, va a recibir una variable llamado *link* que lo pasaremos por el data.

Ahora podemos pasar a la ruta:

```
1. // web.php
2.
3. Route::get('enviar', function(){
4.     $data = [
5.         'link' => 'https://jesuschicano.es'
6.     ];
7.
8.     \Mail::send('emails.notificacion', $data, function($msg){
9.         $msg->from('jesus@shivaweb.es', 'Jesús chicano');
10.        $msg->to('user@example.com')->subject('Notificación');
11.    });
12. }
```

13. );

Lo primero que declaramos en nuestra función anónima es el array *\$data*, el cual contiene un elemento llamado *link*. En este array podemos agregar todos aquellos datos que nos sean de utilidad para incluir en el cuerpo del correo.

Después hacemos la llamada a la clase **Mail** que recibe la vista que habíamos creado antes, el array *\$data* y la función anónima. Esta función recibe un objeto al que se le pasan los datos del *from* y el *to*.

El primero será para el correo origen desde el que se enviaría, y el segundo para el destinatario.

Con esto ya hemos desarrollado un envío de mail cada vez que ingresemos la dirección [www.nuestraaplicacion.com/enviar](http://www.nuestraaplicacion.com/enviar). Si comprobamos la bandeja de entrada de Mailtrap, veremos que ya tenemos un nuevo correo electrónico en el panel derecho y que si lo abrimos, aparece el cuerpo que hemos creado en la vista.

Hay que decir que este tutorial es muy básico y que en futuras ediciones iré actualizando con mayor calidad y mejores herramientas.

## 📁 Laravel, Manual

< Componentes básicos en Vue.js

> Error: Failed to listen on localhost:8000 (Address is already in use)

## Deja un comentario

Publicar comentario

## Categorías

## Entradas recientes

[Instalar certificado SSL gratuitamente mediante FTP](#)

[Cómo cambiar las URLs HTTP a HTTPS en WordPress](#)

[Error: Failed to listen on localhost:8000 \(Address is already in use\)](#)

[Cómo mandar e-mails desde Laravel](#)

[Componentes básicos en Vue.js](#)

