

Cours Laravel - Classe Mail

Ce support présente le fichier fourni concernant l'envoi de mails avec une classe Mail dans Laravel.

```
<?php

// https://laravel.com/docs/11.x/mail           Documentation
// https://laravel.com/docs/11.x/notifications  Notifications
// https://laravel.com/docs/11.x/queues         Files d'attente

// https://symfony.com/doc/7.0/mailer.html      Symfony Mailer (Laravel se sert de cet outil pour les mails)

// Création d'un controller
# php artisan make:controller ContactController

// Création d'une methode index dans le controller

// Faire une vue avec un formulaire de contact

// Faire la route et rajouter un lien dans le menu pour accéder à la page contact

// Créer une class Mail
# php artisan make:mail ContactMail

namespace App\Mail;

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

class ContactMail extends Mailable
{
    use Queueable, SerializesModels;

    public $details; // à rajouter (représente les données provenant du form)

    /**
     * Create a new message instance.
     */
    public function __construct($details) // on rajoute l'argument
    {
        $this->details = $details; // on affecte les données du form dans notre propriété
    }

    /**
     * Get the message envelope.
     */
    public function envelope(): Envelope
    {
        return new Envelope(
            subject: 'Nouvelle demande de contact', // sujet libre
        );
    }

    /**
     * Get the message content definition.
     */
    public function content(): Content
    {
        return new Content(
            view: 'emails.contact', // on précise la vue du template du mail (à créer)
            with: ['details' => $this->details], // On rajoute cette ligne pour rendre disponible dans la vue
        );
    }

    /**
     * Get the attachments for the message.
     *
     * @return array<int, \Illuminate\Mail\Mailables\Attachment>
     */
}
```

```

        public function attachments(): array
        {
            return [];
        }
    }

    // On crée le template du mail par exemple resources/views/emails/contact.blade.php

    <!DOCTYPE html>
    <html lang="fr">
    <head>
        <title>Contact</title>
    </head>
    <body>
        <h2>Nouveau message de contact</h2>
        <p><strong>Nom : </strong>{{ $details['name'] }}</p>
        <p><strong>Email : </strong>{{ $details['email'] }}</p>
        <p><strong>Message : </strong>{{ $details['message'] }}</p>
    </body>
    </html>

    // On met en place la route de validation du form à rajouter dans le action du form et la methode controller

    # Route::post('/send-mail', [ContactController::class, 'sendMail'])->name('send.mail');
    // action="{{ route('send.mail') }}"

    use Illuminate\Support\Facades\Mail; // à rajouter sur le controller (outil mail de laravel)
    use App\Mail\ContactMail; // à rajouter sur le controller (notre class)

    public function sendMail(Request $request)
    {
        $details = [
            'name' => $request->input('name'),
            'email' => $request->input('email'),
            'message' => $request->input('message'),
        ];

        Mail::to('admin@mail.fr')->send(new ContactMail($details));

        return back()->with('success', 'Nous avons reçu votre demande de contact');
    }

    // Pour tester en local par exemple l'outil mailHog (voir le fichier mail_en_local.txt)
    // Si test avec MailHog il faut changer le .env
    MAIL_MAILER=smtp
    MAIL_SCHEME=null
    MAIL_HOST=127.0.0.1
    MAIL_PORT=1025
    MAIL_USERNAME=null
    MAIL_PASSWORD=null
    MAIL_FROM_ADDRESS="hello@example.com"
    MAIL_FROM_NAME="Laravel Dashboard"

```