

# Laravel et MDL

Material Design est un langage visuel mis au point par Google. Concernant la partie Material elle évoque le fait que le visuel se rapproche de la réalité matérielle : le papier, l'encre, les ombres... L'initiative avait pour objectif d'unifier leurs applications et de les rendre homogènes sur tous les supports.

Le Material Design c'est essentiellement des règles de design avec des formes simple et lisibles et des effets visuels pour renseigner l'utilisateur de ses actions. On trouve toutes les règles et explications [sur le site concerné](#). C'est une lecture très saine et instructive !

Google nous offre de nombreux outils pour mettre en œuvre le Material Design mais le plus pratique est certainement [Materiel Design Lite](#) (MDL). C'est une librairie CSS et Javascript légère et facile à utiliser.

Je me suis dit qu'il serait intéressant de la mettre en œuvre avec Laravel.

Par défaut Laravel arrive avec Bootstrap et JQuery mais on n'est évidemment pas obligés de les utiliser. Dans cet article je montre comment transformer l'installation de base pour utiliser MDL.

## Installation de Laravel

Si comme moi vous utilisez [Laragon](#) il est facile de créer rapidement une installation toute neuve de Laravel :



© Leo K

## Menu

Apache httpd-2.4.35-win64-VC15 Démarré

80

MySQL mysql-5.7.24-winx64 Démarré

3306

Créer un site web rapidement

Nom du Projet [Laravel]:

laravel-mdl

OK Cancel

Si tout se passe bien vous devez obtenir la page d'accueil :

Sinon vous devez taper vous-même les commandes :

```
composer create-project laravel/laravel laravel-mdl --prefer-dist
```

Si tout se passe bien vous devez obtenir la page d'accueil :

# Laravel

[DOCUMENTATION](#)

[LARACASTS](#)

[NEWS](#)

[NOVA](#)

[FORGE](#)

[GITHUB](#)

0

On va ajouter les éléments pour l'authentification :

```
php artisan make:auth
```

On va aussi créer une base (Laragon s'en charge aussi). On renseigne le fichier `.env` pour pointer sur la bonne base :

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel-mdl
DB_USERNAME=root
DB_PASSWORD=
```

Et on lance les migrations par défaut :

```
php artisan migrate
```

On va aussi charger les librairies avec npm :

```
npm install
```

Voilà on a maintenant un Laravel tout neuf qu'on va pouvoir triturer...

## Material Design Lite

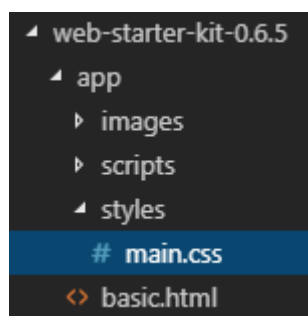
On trouve tous les renseignements sur [ce site](#). Pour mettre en œuvre MDL il suffit de faire appel aux librairies :

```
<link                                rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
<link                                rel="stylesheet"
href="https://code.getmdl.io/1.3.0/material.indigo-pink.min.css">
<script                               defer
src="https://code.getmdl.io/1.3.0/material.min.js"></script>
```

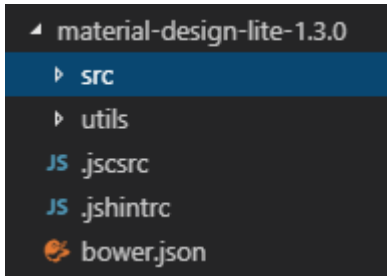
Dans l'appel à la librairie CSS on choisit le thème coloré désiré. Et c'est tout !

Mais pour rendre le CSS plus facile à modifier pour nos besoins il vaut mieux utiliser la version **Sass** et la compiler avec **mix**. Pour utiliser la version Sass on nous explique la procédure [ici](#). On ne va pas la suivre exactement pour Laravel.

Il faut télécharger le [Web Starter Guide](#). Le seul fichier qui va nous intéresser est **main.css** :



Ensuite on télécharge [Material Design Lite](#). Là on va récupérer tout ce qui se trouve dans le dossier **src** :



Dans l'installation de Laravel on a un dossier avec des fichiers Sass :

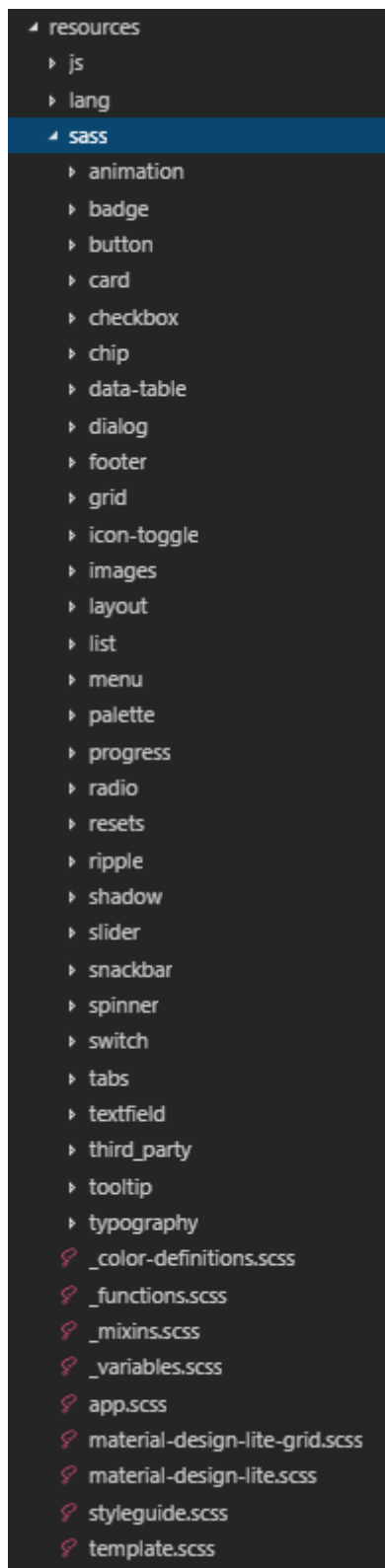


On supprime ces deux fichiers et on copie tout le contenu de **material-design-lite-1.3.0/src** (en ne conservant que les fichiers **scss**) ainsi que le fichier **web-starter-kit-0.6.5/app/styles/main.css**. Changez le nom de ce dernier fichier en **app.scss**. Changez le code de ce fichier pour celui-ci :

```
@import "material-design-lite";
```

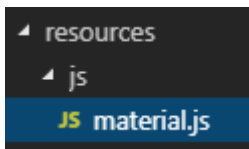
```
html, body {  
  Roboto', 'Helvetica', sans-serif;  
  margin: 0;  
  padding: 0;  
}
```

Vous devez vous retrouver finalement avec ça :



Au niveau du Javascript par défaut Laravel prévoit Bootstrap, JQuery, lodash, popper, et axios. On va se passer de tout ça en se contentant de la librairie Javascript de MDL.

Supprimez tout ce qui se trouve dans **resources/js** et copiez le fichier **material-design-lite-1.3.0/material.js** :



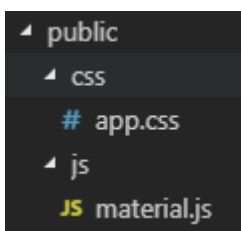
Dans le fichier **webpack.mix.js** changez ainsi le code :

```
mix.js('resources/js/material.js', 'public/js')  
    .sass('resources/sass/app.scss', 'public/css');
```

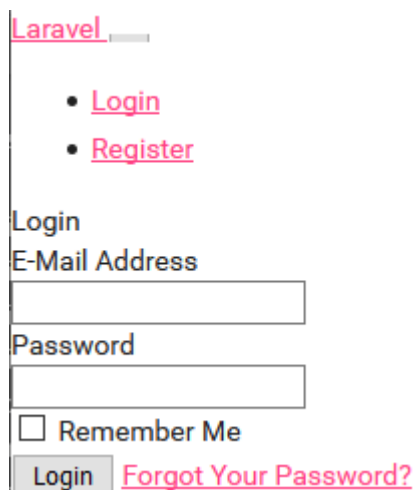
Il ne reste plus qu'à relancer la compilation en mode développement :

```
npm run dev
```

Normalement vous avez un fichier **public/css/app.css** tout neuf ainsi qu'un nouveau fichier **public/js/material.js** (supprimez l'ancien app.js) :



Mais évidemment si la page d'accueil de Laravel demeure inchangée parce qu'indépendante du CSS global il n'en va pas de même de toutes les autres vues, par exemple pour le login :



Il faut maintenant reprendre toutes les vues pour MDL...

# Le layout

On va commencer par mettre à jour le layout (`resources/views/layouts/app.blade.php`) :

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>{{ config('app.name', 'Laravel') }}</title>

    <!-- Fonts -->
                                <link                rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">

    <!-- Styles -->
    <link href="{{ asset('css/app.css') }}" rel="stylesheet">

    <!-- Scripts -->
    <script src="{{ asset('js/material.js') }}" defer></script>

</head>
<body>

    <div class="mdl-layout mdl-js-layout mdl-layout--fixed-
header">
        <header class="mdl-layout__header">
            <div class="mdl-layout__header-row">
                <!-- Title -->
                <span class="mdl-layout-title"><a href="{{ url('/')
}}">{{ config('app.name', 'Laravel') }}</a></span>
                <!-- Add spacer, to align navigation to the right -->
                <div class="mdl-layout-spacer"></div>
                <!-- Navigation. We hide it in small screens. -->
                <nav class="mdl-navigation mdl-layout--large-screen-
only">
```

```

        <!-- Authentication Links -->
        @guest
            <a class="mdl-navigation__link" href="{{
route('login') }}">{{ __('Login') }}</a>
            @if (Route::has('register'))
                <a class="mdl-navigation__link" href="{{
route('register') }}">{{ __('Register') }}</a>
            @endif
        @else
            <button class="mdl-button mdl-js-button"
id="logout">
                {{ Auth::user()->name }}
            </button>
            <ul class="mdl-menu mdl-js-menu mdl-js-ripple-
effect mdl-menu--bottom-right" for="logout">
                <li class="mdl-menu__item"
                    onclick="event.preventDefault();
                    document.getElementById('logout-
form').submit();">
                    {{ __('Logout') }}
                </li>
            </ul>
            <form id="logout-form" action="{{
route('logout') }}" method="POST" style="display: none;">
                @csrf
            </form>
        @endguest
    </nav>
</div>
</header>
<div class="mdl-layout__drawer">
    <span class="mdl-layout-title"><a href="{{ url('/')
}}">{{ config('app.name', 'Laravel') }}</a></span>
    <nav class="mdl-navigation">

        <!-- Authentication Links -->
        @guest
            <a class="mdl-navigation__link" href="{{
route('login') }}">{{ __('Login') }}</a>
            @if (Route::has('register'))
                <a class="mdl-navigation__link" href="{{
route('register') }}">{{ __('Register') }}</a>
            @endif

```



```

        @else
            <a class="mdl-navigation__link" href="{{
route('logout') }}"
                onclick="event.preventDefault();
                document.getElementById('logout-
form').submit();">
                {{ __('Logout') }}
            </a>
        @endguest

    </nav>
</div>

@yield('content')

</div>

</body>
</html>

```

On utilise [le composant layout de MDL](#). Il permet de créer une barre de navigation :



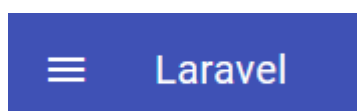
On a juste un petit souci avec le titre Laravel qu'il faut un peu arranger. On ajoute ces règles dans **resources/sass/app.scss** :

```

.mdl-layout-title {
  a {
    color: inherit;
  }
  a:link {
    text-decoration: none;
  }
}

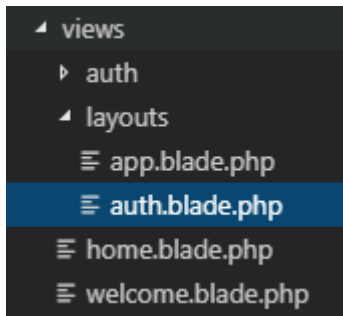
```

Relancez la compilation (ou mettez en mode **watch** pour que ce soit automatique).



Maintenant notre barre de navigation est parfaite.

On va créer un layout spécifique pour la partie authentication parce qu'on veut un centrage correct des formulaires, ce qui ne sera pas forcément le cas pour le reste du site :



Avec ce code :

```
@extends('layouts.app')
```

```
@section('content')
```

```
<div class="mdl-layout" id="main_layout">
  <main class="mdl-layout__content">
    <div class="page-content">
      @yield('authContent')
    </div>
  </main>
</div>
```

```
@endsection
```

Et ces règles dans **resources/sass/app.scss** :

```
#main_layout {

  align-items: center;
  justify-content: center;

  .mdl-layout__content {
    padding: 24px;
    flex: none;
  }

  .mdl-textfield__error {
    visibility: visible;
  }
}
```

```
}  
}
```

# Le login

On va maintenant changer le code pour la vue de login (**resources/views/auth/login.blade.php**) :

```
@extends('layouts.auth')
```

```
@section('authContent')
```

```
    <div class="mdl-card mdl-shadow--6dp">  
        <div class="mdl-card__title mdl-color--primary mdl-color-text--white">  
            <h2 class="mdl-card__title-text">{{ __('Login') }}</h2>  
        </div>  
        <div class="mdl-card__supporting-text">  
            <form method="POST" action="{{ route('login') }}">  
                @csrf  
  
                <div class="mdl-textfield mdl-js-textfield mdl-textfield--floating-label">  
                    <input class="mdl-textfield__input" type="email" id="email" name="email" value="{{ old('email') }}" required autofocus>  
                    <label class="mdl-textfield__label" for="email">{{ __('E-Mail Address') }}</label>  
                    @if($errors->has('email'))  
                        <span class="mdl-textfield__error">{{ $errors->first('email') }}</span>  
                    @endif  
                </div>  
  
                <div class="mdl-textfield mdl-js-textfield mdl-textfield--floating-label">  
                    <input class="mdl-textfield__input" type="password" id="password" name="password" required>  
                    <label class="mdl-textfield__label" for="password">{{ __('Password') }}</label>  
                    @if($errors->has('password'))
```

```

                <span class="mdl-textfield__error">{{
$errors->first('password') }}</span>
            @endif
        </div>

        <label class="mdl-switch mdl-js-switch mdl-js-ripple-effect" for="remember">
            <input type="checkbox" id="remember"
class="mdl-switch__input" {{ old('remember') ? 'checked' : '' }}>
                <span class="mdl-switch__label">{{
__( 'Remember Me' ) }}</span>
            </label>

            <button id="action-button" type="submit"
class="mdl-button mdl-js-button mdl-button--raised mdl-js-ripple-effect mdl-button--colored mdl-color-text--white">
                {{ __( 'Login' ) }}
            </button>

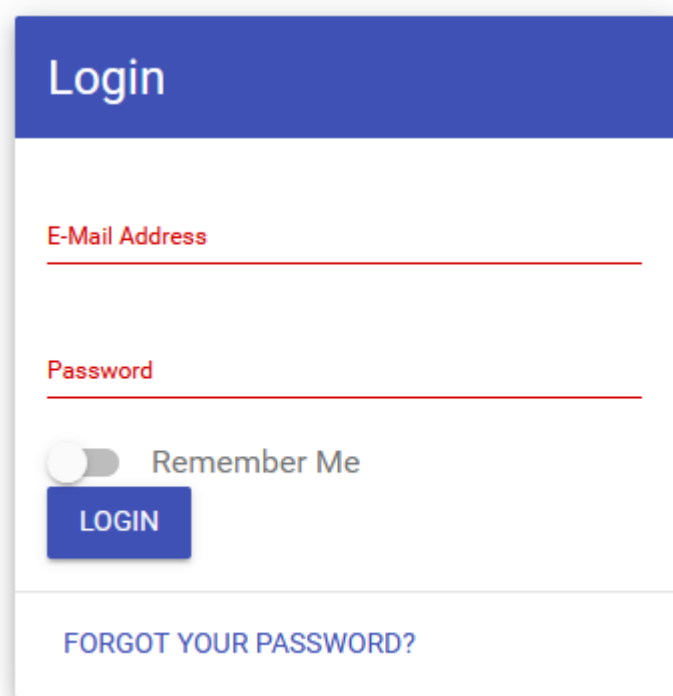
        </form>
    </div>
    <div class="mdl-card__actions mdl-card--border">
        @if (Route::has('password.request'))
            <a class="mdl-button mdl-button--colored mdl-js-button mdl-js-ripple-effect" href="{{ route('password.request') }}">
                {{ __( 'Forgot Your Password?' ) }}
            </a>
        @endif
    </div>
</div>

```

@endsection

On utilise [le composant Cards de MDL](#). D'autre part dans le formulaire on utilise [le composant Text Fields](#).

Voilà le résultat bien centré dans la page :

A login form with a blue header containing the word "Login". Below the header are two input fields: "E-Mail Address" and "Password", each with a red underline. Below the "Password" field is a "Remember Me" toggle switch. A blue "LOGIN" button is positioned below the toggle. At the bottom of the form is a link that says "FORGOT YOUR PASSWORD?".

Login

E-Mail Address

Password

Remember Me

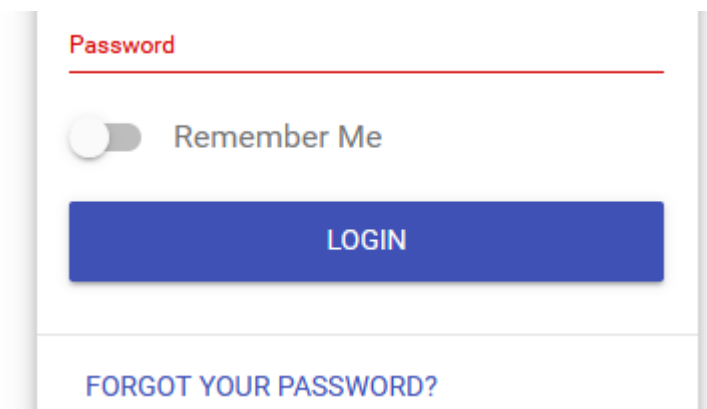
LOGIN

FORGOT YOUR PASSWORD?

On va juste arranger un peu le

bouton :

```
#action-button {  
  width: 100%;  
  height: 40px;  
  min-width: initial;  
  margin: 20px 0 10px;  
}
```

The same login form as above, but with the "LOGIN" button now stretched to 100% width. The "Remember Me" toggle and "FORGOT YOUR PASSWORD?" link remain in their original positions.

Password

Remember Me

LOGIN

FORGOT YOUR PASSWORD?

Maintenant c'est parfait !

On vérifie que les messages d'erreurs s'affichent bien :

# L'enregistrement

On va faire la même chose pour la vue `resources/views/auth/register.blade.php` :

```
@extends('layouts.auth')
```

```
@section('authContent')
```

```
    <div class="mdl-card mdl-shadow--6dp">
        <div class="mdl-card__title mdl-color--primary mdl-color-
text--white">
            <h2 class="mdl-card__title-text">{{ __('Register')
}}</h2>
        </div>
        <div class="mdl-card__supporting-text">
            <form method="POST" action="{{ route('register') }}">
                @csrf

                <div class="mdl-textfield mdl-js-textfield mdl-
textfield--floating-label">
                    <input class="mdl-textfield__input"
type="text" id="email" name="name" value="{{ old('name') }}"
required autofocus>
                    <label class="mdl-textfield__label"
for="name">{{ __('Name') }}</label>
                    @if($errors->has('name'))
                        <span class="mdl-textfield__error">{{
$errors->first('name') }}</span>
                    @endif
                </div>

                <div class="mdl-textfield mdl-js-textfield mdl-
textfield--floating-label">
                    <input class="mdl-textfield__input"
type="email" id="email" name="email" value="{{ old('email') }}"
required autofocus>
                    <label class="mdl-textfield__label"
for="email">{{ __('E-Mail Address') }}</label>
                    @if($errors->has('email'))
```

```

                <span class="mdl-textfield__error">{{
$errors->first('email') }}</span>
            @endif
        </div>

        <div class="mdl-textfield mdl-js-textfield mdl-
textfield--floating-label">
            <input class="mdl-textfield__input"
type="password" id="password" name="password" required>
                <label class="mdl-textfield__label"
for="password">{{ __( 'Password' ) }}</label>
                @if($errors->has('password'))
                    <span class="mdl-textfield__error">{{
$errors->first('password') }}</span>
                @endif
            </div>

            <div class="mdl-textfield mdl-js-textfield mdl-
textfield--floating-label">
                <input class="mdl-textfield__input"
type="password"
                    id="password_confirmation"
name="password_confirmation" required>
                    <label class="mdl-textfield__label"
for="password_confirmation">{{ __( 'Confirm Password' ) }}</label>
                </div>

                <button id="action-button" type="submit"
class="mdl-button mdl-js-button mdl-button--raised mdl-js-ripple-
effect mdl-button--colored mdl-color-text--white">
                    {{ __( 'Register' ) }}
                </button>

        </form>
    </div>
</div>

@endsection

```

# Register

Name  
\_\_\_\_\_

E-Mail Address  
\_\_\_\_\_

Password  
\_\_\_\_\_

Confirm Password  
\_\_\_\_\_

REGISTER

On en profite pour créer un user :

Name  
Durand  
\_\_\_\_\_

E-Mail Address  
durand@moi.fr  
\_\_\_\_\_

Password  
●●●●●●●●  
\_\_\_\_\_

Confirm Password  
●●●●●●●●  
\_\_\_\_\_

On y arrive bien mais on a un petit souci avec la vue lorsqu'on est connecté.

Le nom apparaît bien mais il faudrait qu'il soit en blanc :



```
#logout {
    color: white;
}
```

Ensuite on change le code de la vue **resources/views/home.blade.php** :

```
@extends('layouts.auth')
```

```
@section('authContent')
```

```
    <div class="mdl-card mdl-shadow--6dp">
        <div class="mdl-card__title mdl-color--primary mdl-color-
text--white">
            <h2 class="mdl-card__title-text">Dashboard</h2>
        </div>

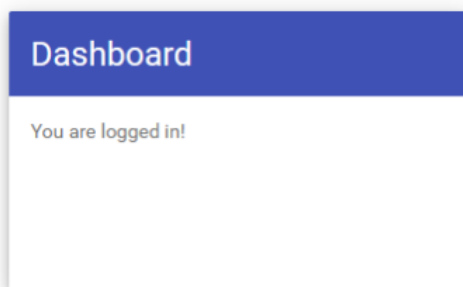
        <div class="mdl-card__supporting-text">

            @if (session('status'))
                <div class="mdl-card__supporting-text mdl-color-
text--primary">
                    {{ session('status') }}
                </div>
                <hr>
            @endif
            You are logged in!

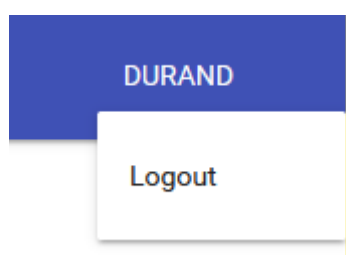
        </div>
    </div>
```

```
@endsection
```

C'est maintenant plus présentable :



Pour le logout on a un menu déroulant :



Il est ajouté à la barre de navigation avec [le composant Menus](#).

## Le renouvellement du mot de passe

Pour le renouvellement du mot de passe on met à jour la vue `resources/views/auth/passwords/email.blade.php` :

```
@extends('layouts.auth')
```

```
@section('authContent')
```

```
    <div class="mdl-card mdl-shadow--6dp">
        <div class="mdl-card__title mdl-color--primary mdl-color-text--white">
            <h2 class="mdl-card__title-text">{{ __('Reset Password') }}</h2>
        </div>
```

```
    <div class="mdl-card__supporting-text">
        <form method="POST" action="{{ route('password.email') }}
```

```

}}">
        @csrf

        @if (session('status'))
            <div class="mdl-card__supporting-text mdl-
color-text--primary">
                {{ session('status') }}
            </div>
            <hr>
        @endif

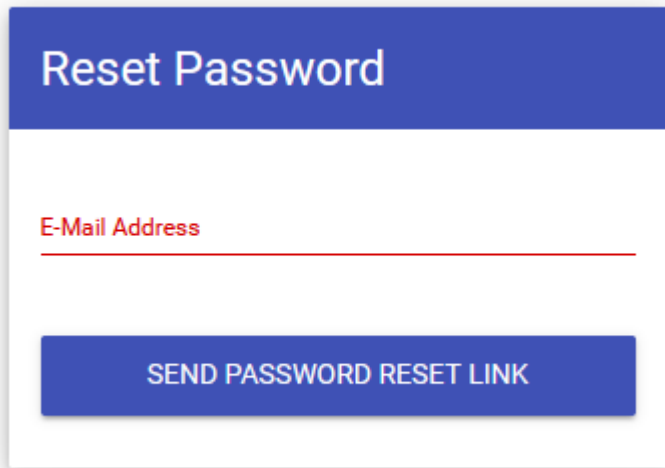
        <div class="mdl-textfield mdl-js-textfield mdl-
textfield--floating-label">
            <input class="mdl-textfield__input"
type="email" id="email" name="email" value="{{ old('email') }}"
required autofocus>
                <label class="mdl-textfield__label"
for="email">{{ __( 'E-Mail Address' ) }}</label>
                @if($errors->has('email'))
                    <span class="mdl-textfield__error">{{
$errors->first('email') }}</span>
                @endif
            </div>

            <button id="action-button" type="submit"
class="mdl-button mdl-js-button mdl-button--raised mdl-js-ripple-
effect mdl-button--colored mdl-color-text--white">
                {{ __( 'Send Password Reset Link' ) }}
            </button>

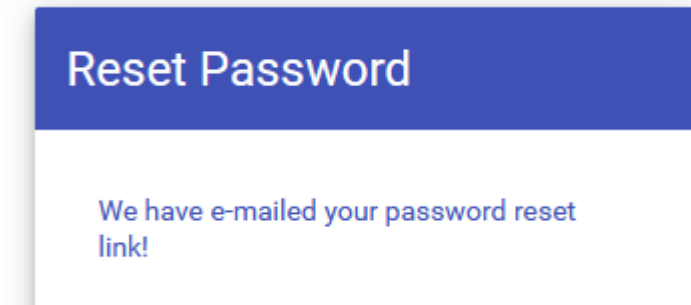
        </form>
    </div>
</div>

@endsection

```



On vérifie que l'email part bien et qu'on a le message :



On change aussi le code de la vue pour modifier le mot de passe (**resources/views/auth/passwords/reset.blade.php**) :

```
@extends('layouts.auth')
```

```
@section('authContent')
```

```
    <div class="mdl-card mdl-shadow--6dp">
        <div class="mdl-card__title mdl-color--primary mdl-color-
text--white">
            <h2 class="mdl-card__title-text">{{ __('Reset
Password') }}</h2>
        </div>
        <div class="mdl-card__supporting-text">
            <form method="POST" action="{{
route('password.update') }}">
                @csrf

                <input type="hidden" name="token" value="{{ $token
}}">
```

```
        <div class="mdl-textfield mdl-js-textfield mdl-
textfield--floating-label">
            <input class="mdl-textfield__input"
type="email" id="email" name="email" value="{{ old('email') }}"
required autofocus>
                <label class="mdl-textfield__label"
for="email">{{ __( 'E-Mail Address' ) }}</label>
                    @if($errors->has('email'))
                        <span class="mdl-textfield__error">{{
$errors->first('email') }}</span>
                    @endif
            </div>
```

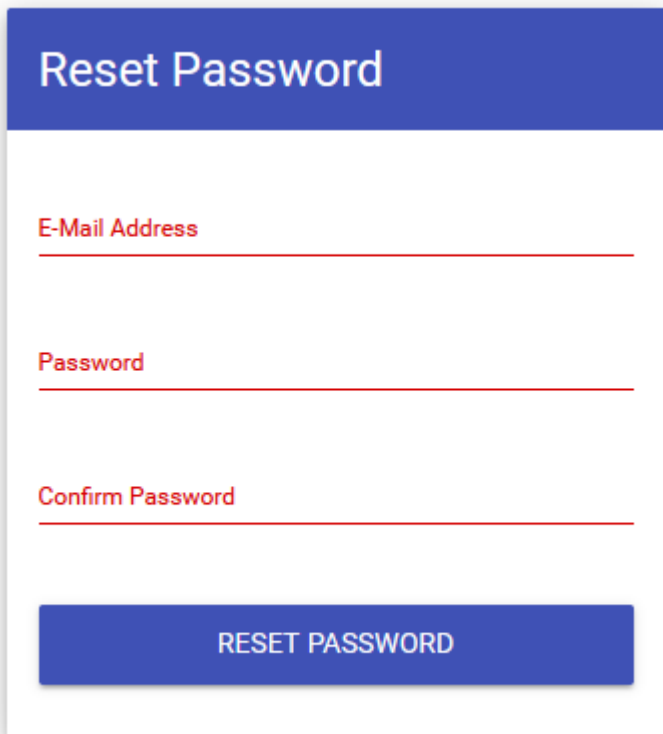
```
        <div class="mdl-textfield mdl-js-textfield mdl-
textfield--floating-label">
            <input class="mdl-textfield__input"
type="password" id="password" name="password" required>
                <label class="mdl-textfield__label"
for="password">{{ __( 'Password' ) }}</label>
                    @if($errors->has('password'))
                        <span class="mdl-textfield__error">{{
$errors->first('password') }}</span>
                    @endif
            </div>
```

```
        <div class="mdl-textfield mdl-js-textfield mdl-
textfield--floating-label">
            <input class="mdl-textfield__input"
type="password"
            id="password_confirmation"
name="password_confirmation" required>
                <label class="mdl-textfield__label"
for="password_confirmation">{{ __( 'Confirm Password' ) }}</label>
            </div>
```

```
        <button id="action-button" type="submit"
class="mdl-button mdl-js-button mdl-button--raised mdl-js-ripple-
effect mdl-button--colored mdl-color-text--white">
            {{ __( 'Reset Password' ) }}
        </button>
```

```
    </form>
</div>
</div>
```

@endsection



Parfait !

## La vérification de l'email

Laravel offre la possibilité de faire une vérification de l'email lorsque quelqu'un s'enregistre. Il faut le préciser dans les routes de l'authentification :

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

Ensuite on définit les routes à protéger, par exemple celle de la vue **home** :

```
Route::get('/home',  
'HomeController@index')->name('home')->middleware('verified');
```

Enfin il faut dans le modèle User implémenter l'interface :

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

On va aussi adapter la vue **resources/views/auth/verify.blade.php** pour MDL :

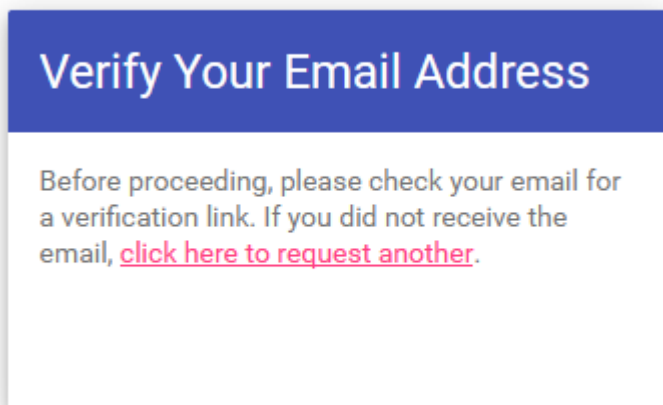
```
@extends('layouts.auth')
```

```
@section('authContent')
```

```
    <div class="mdl-card mdl-shadow--6dp">
        <div class="mdl-card__title mdl-color--primary mdl-color-
text--white">
            <h2 class="mdl-card__title-text">{{ __('Verify Your
Email Address') }}</h2>
        </div>

        <div class="mdl-card__supporting-text">
            {{ __('Before proceeding, please check your email for
a verification link.') }}
            {{ __('If you did not receive the email') }}, <a
href="{{ route('verification.resend') }}">{{ __('click here to
request another') }}</a>.
        </div>
    </div>
```

```
@endsection
```



aussi !

Et c'est bon pour cette partie

## Conclusion

On a vu qu'il n'est pas si difficile d'utiliser MDL avec Laravel. Les fichiers générés en production sont très légers :

Asset	Size	Chunks		Chunk Names
/css/app.css	125 KiB	0	[emitted]	/js/material
/js/material.js	61.8 KiB	0	[emitted]	/js/material

D'autre part on dispose de tous les composants nécessaires pour construire un site plutôt esthétique. Évidemment il faudra ensuite certainement ajouter une librairie Javascript pour gérer efficacement le client, il serait alors dommage d'utiliser JQuery. On peut bien sûr opter pour Vue.js mais aussi se contenter d'une librairie légère comme [RE:DOM](#).

Pour ceux qui n'arriveraient pas à construire l'application ou trop fainéants pour tout faire voici [un zip du projet final](#).