Cours Laravel 6 – les données – la vérification de l'email

laravel.sillo.org/cours-laravel-6-les-donnees-la-verification-de-lemail/

bestmomo

August 31, 2019

Pour enregistrer un nouvel utilisateur on a vu qu'on demande une adresse email. C'est une procédure très classique et assez généralisée. Mais il se peut que l'adresse fournie soit totalement fantaisiste. Ça peut selon les cas être sans conséquence ou plus gênant. Laravel est équipé pour la vérification de la validité de cet email.

Modèle et base

Pour que la vérification de l'email soit effective il faut faire une modification dans le code du modèle **User**. Il doit utiliser l'interface **MustVerifyEmail** :

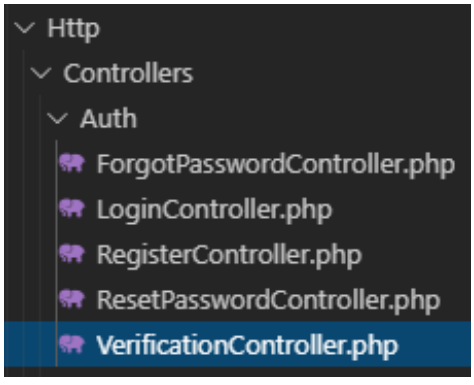
```
use Illuminate\Contracts\Auth\MustVerifyEmail;
...

class User extends Authenticatable implements MustVerifyEmail
```

D'autre part la table **users** doit comporter une colonne **email_verified_at**. Et il se trouve que cette colonne est déjà prévue dans la migration de base et a donc été créée :

Name	Data Type
id	BIGINT(20)
name	VARCHAR(255)
email	VARCHAR(255)
email_verified_at	TIMESTAMP
password	VARCHAR(255)
remember_token	VARCHAR(100)
created_at	TIMESTAMP
updated_at	TIMESTAMP

Routes et contrôleur

Si vous regardez dans les contrôleurs vous allez voir que lorsque nous avons mis en place l'authentification il s'est créé le contrôleur **VerificationController** :



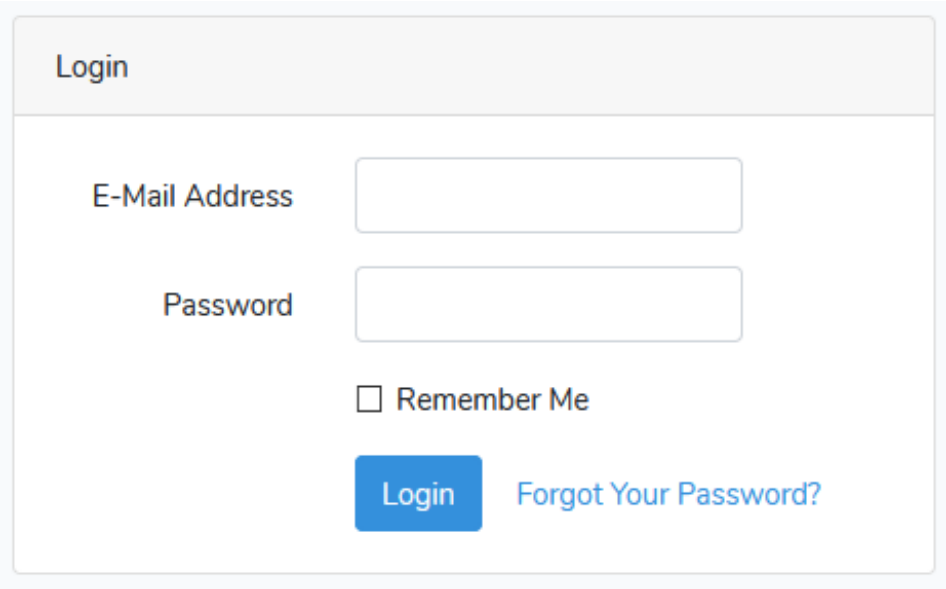
Pour que ce contrôleur serve à quelque chose on va compléter la déclaration des routes de l'authentification ainsi :

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

Laravel est équipé de base du middleware **verified** qu'on peut placer sur les routes qu'on veut rendre inaccessibles tant que l'email n'a pas été confirmé :

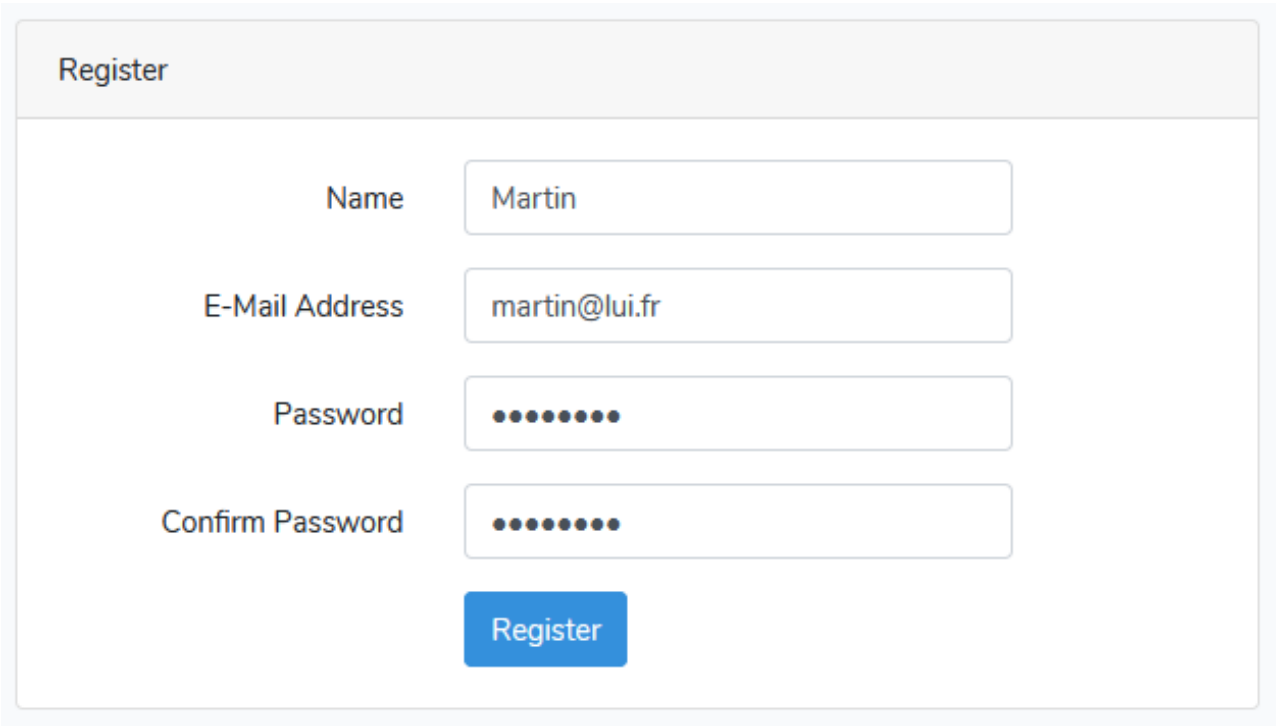
```
Route::get('protege', function () {
    return 'affichage de la route protégé';
})->middleware('verified');
```

Que se passe-t-il quand on essaie d'accéder à cette route si on n'est pas connecté ? Tout simplement on se retrouve avec le formulaire de connexion :

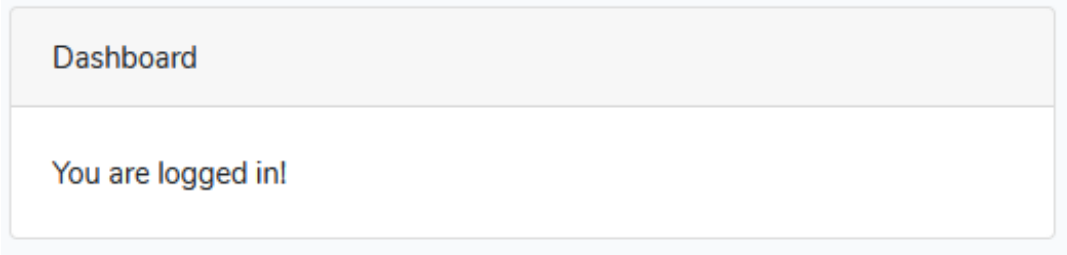


Le processus

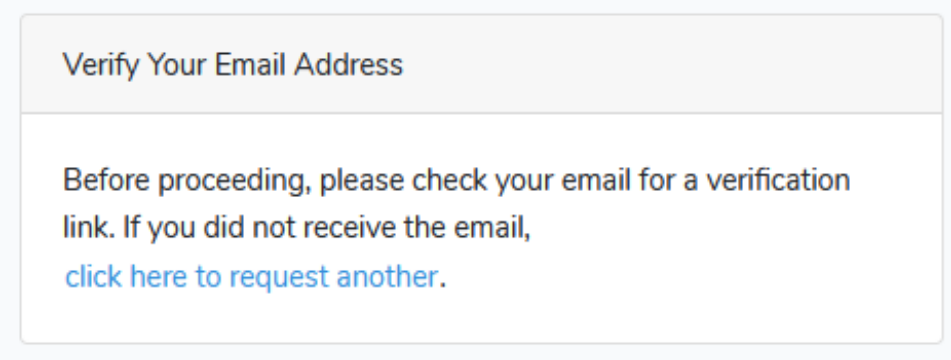
On va donc inscrire un nouvel utilisateur :



De façon classique il arrive sur la page qui lui dit que tout va bien et qu'il est connecté :

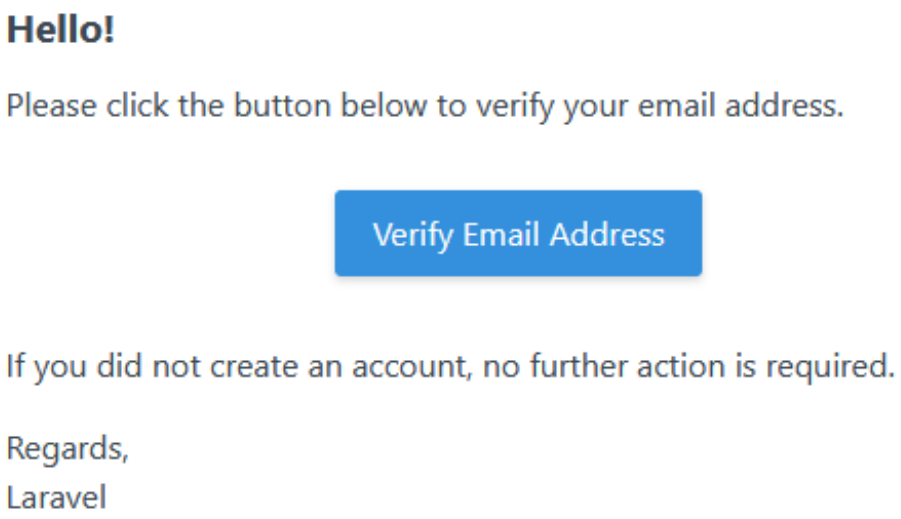


Est-ce bien logique ? Je n'ai pas prévu de protéger la route correspondante alors on va dire que oui. Par contre si je vais sur la route que j'ai protégée :



On voit que cette fois il faut confirmer l'adresse pour avoir l'accès.

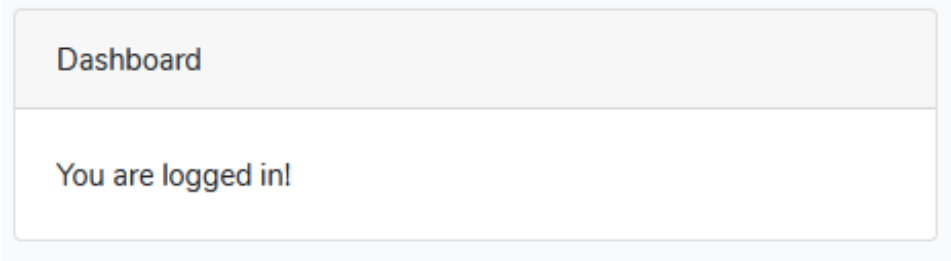
L'utilisateur a normalement reçu cet email :



Si on clique sur le bouton Laravel va renseigner la colonne dans la table :

Et maintenant l'utilisateur est connecté et peut accéder aux routes protégées. Par défaut il arrive sur la page **home** :

name	email	email_verified_at
Martin	martin@lui.fr	2019-08-31 11:54:30



La redirection est définie dans le contrôleur **VerificationController** :

```
protected $redirectTo = '/home';
```

Elle est donc facile à changer.

En résumé

- Lorsqu'on ajoute le package **laravel/ui** on ajoute en même temps la possibilité de vérifier l'email d'un nouvel inscrit.
- On peut protéger les routes sensibles avec le middleware **verified** tant que l'email n'est pas confirmé.