Pagination Laravel Livewire

1. Objectifs

- Etre capable d'ajouter une simple pagination livewire dans une application laravel 8.

2. Présentation

- Livewire est un framework full-stack pour Laravel qui simplifie la création d'interfaces dynamiques, sans quitter le confort de Laravel.
- Laravel Livewire offre la possibilité de paginer les résultats dans un composant. Cette fonctionnalité se connecte aux fonctionnalités de pagination natives de Laravel, elle devrait donc vous sembler une fonctionnalité invisible.
- Ce tutoriel se concentrer sur l'exemple de pagination de laravel 8 livewire.
- Noter que nous pouvons le faire en utilisant Ajax, Vue ou React. Mais dans cet exemple, nous allons créer le même personnel en utilisant les packages Livewire. Cela nous donnera un système de pagination très léger dans notre application.

3. Réalisation

1. Créer un nouveau projet Laravel8

- Dans la première étape, nous devons obtenir une nouvelle application de version laravel à l'aide de la commande ci-dessous. Alors ouvrons le terminal et exécutons la commande ci-dessous.

```
composer create-project --prefer-dist laravel/laravel livewire-liste-dependante
```

2. Configurer la base de données

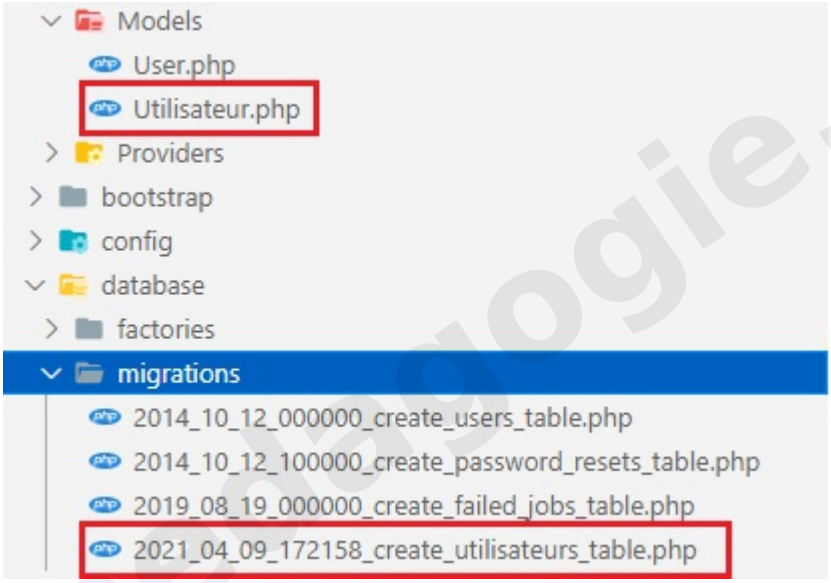
- Pour configurer la base de données. Ouvrir le fichier "`.env`"
- changer le nom de la base de données (livewire01), le nom d'utilisateur et le mot de passe.

apcpedagogie.com



3. **Ajouter un fichier de migration et un fichier modèle**

   - La commande suivante crée deux fichiers : **`php artisan make:model Utilisateur -m`**



   - **Contenu des fichers**

     1. Fichier **database\migrations\....._create_utilisateurs_table.php**

```
public function up()
    {
        Schema::create('utilisateurs', function (Blueprint $table) {
            $table->integer('cin')->primary();
            $table->string('prenomNom');
            $table->date('dteNaissance');
            $table->string('genre');
            $table->string('pays');
            $table->string('email');
```

apcpedagogie.com

```
            $table->timestamps();
        });
    }
```

4. Fichier **App\Models\Utilisateur.php**

```
class Utilisateur extends Model
{
    use HasFactory;
    public $primaryKey = 'cin';
    public $incrementing = false;
    protected $fillable = [
        'cin', 'prenomNom', 'dteNaissance', 'genre', 'pays' , 'email'
    ];
}
```

   - Vous devez maintenant exécuter cette migration en suivant la commande: `php artisan migrate`

4. **Générer un seeder**

   - Pour générer un seeder, exécutez la commande suivant: **`php artisan make:seeder UtilisateursTableSeeder`**

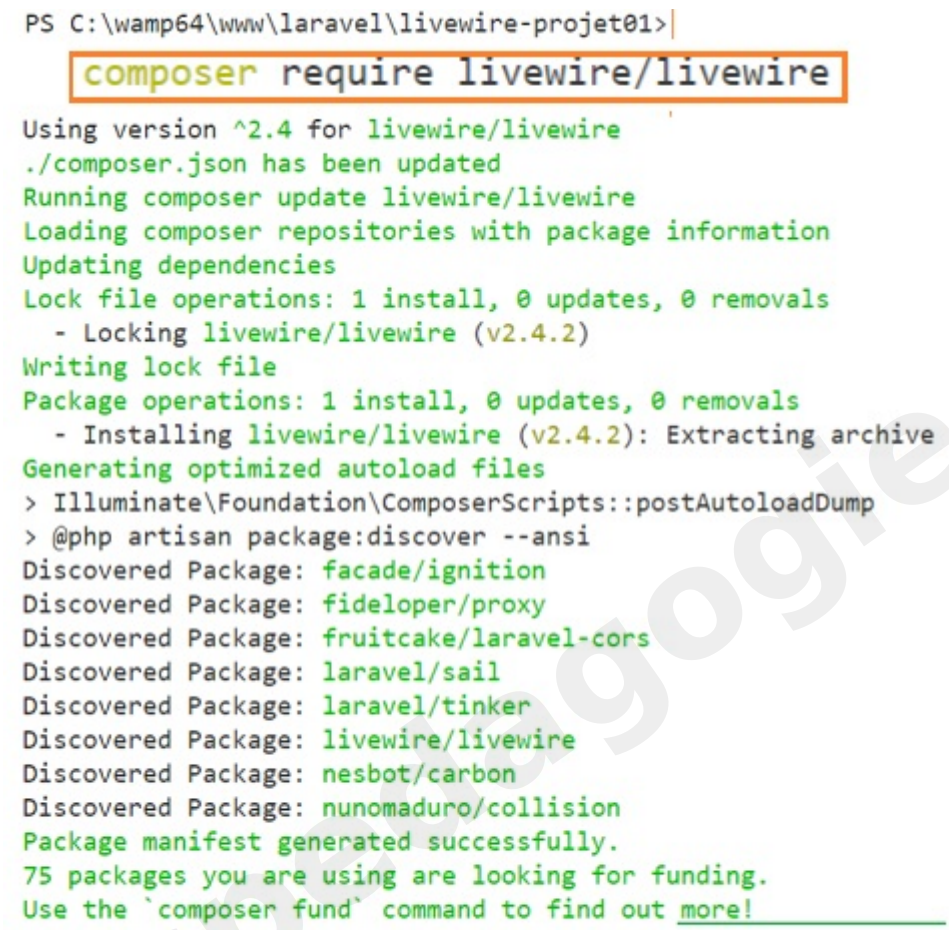


   - Vous pouvez utilisez la commande suivante pour insérer les données ci-dessus dans la table "**utilisateurs**" : `php artisan db:seed --class=UtilisateursTableSeeder`

5. **Installer Livewire**

- **Livewire** est un package pour Laravel qui s'installe classiquement avec composer :
- Dans cette étape, installer **livewire** sur l'application laravel en utilisant la commande ci-dessous:

```
composer require livewire/livewire
```



6. Cette commande installera le package Livewire dans le framework Laravel 8.

7. **Créer Component**

   - Ensuite, nous voulons créer un composant Livewire. Donc, pour cela aussi, nous devons aller à l'invite de commande et écrire la commande suivante.

apcpedagogie.com

```
PS C:\wamp64\www\laravel\livewire-pagination> php artisan make:livewire utilisateurs
COMPONENT CREATED

CLASS: app/Http/Livewire/Utilisateurs.php
VIEW:  C:\wamp64\www\laravel\livewire-pagination\resources\views/livewire/utilisateurs.blade.php

  _-_
 / /o\ \   || ()              () __
|_\ /_|   || || \\// /_\ \\ // || |~~ /_\
 |`|`|    || ||  \/  \\_  \^/  || ||  \\_

Congratulations, you've created your first Livewire component!
```

- Cette commande générera deux fichiers, l'un est la classe **Utilisateurs.php** dans le répertoire **app/Http/Livewire** et le fichier de vues **utilisateurs.blade.php** dans le répertoire **views/livewire**.
- Ces deux fichiers seront utilisés pour créer une recherche Livewire avec une application de pagination.

```
app/Http/Livewire/Utilisateurs.php
resources/views/livewire/utilisateurs.blade.php
```

8. Contenu du ficher: **app/Http/Livewire/Utilisateurs.php**

9. **Mettre à jour le fichier de component**

- Contenu du ficher: **app/Http/Livewire/Utilisateurs.php**

```php
<?php
namespace App\Http\Livewire;
use Livewire\Component;
use App\Models\Utilisateur;
use Livewire\WithPagination;
class Utilisateurs extends Component
{
    use WithPagination;
    protected $paginationTheme = 'bootstrap';
    public function render()
    {
        return view('livewire.utilisateurs', [
            'utilisateurs' => Utilisateur::paginate(4),]);
    }
}
```

- Donc, nous avons d'abord ouvert le fichier **app/Http/Livewire/Utilisateurs.php** et sous ce fichier, nous avons ajouté :

- **à l'en-tête**
  - `use App\Models\Utilisateur;`
  - `use Livewire\WithPagination;`
- **Sous cette classe**
  - `use WithPagination;`
  - `protected $paginationTheme = 'bootstrap';`
- Sous la méthode de **render** de classe **Utilisateurs**, qui est la méthode racine de cette classe, nous avons défini le code pour les données de recherche et créé des liens de pagination.

10. **Ajouter une route**

- Pour ajouter une route dans l'application laravel. ouvrir le fichier "**routes/web.php**" et ajouter la route suivante.

```
Route::view('utilisateurs','livewire.home');
```

11. **Créer les views**

- **resources/views/livewire/utilisateurs.blade.php**
  - Dans cet étape, nous devons aller au fichier **views/livewire/utilisateurs.blade.php** et sous ce fichier, nous avons créé une table **html** et remplissez cette table avec des données et sous cette table, nous avons également fait des liens de pagination.

```
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-10">
            <div class="card">
                <div class="card-header">
                    <div class="row justify-content-evenly">
                            <h4>Laravel Livewire pagination -
apcpedagogie.com</h4>
                    </div>
                </div>
                <div class="card-body">
                    <div>
                        <table class="table table-bordered mt-5">
                            <thead>
                                <tr><th>CIN</th><th>Prénom & Nom</th><th>Date
```

```
Naissance</th>

<th>Genre</th><th>Pays</th><th>Email</th><th>Action</th></tr>
                            </thead>
                            <tbody>
                                @foreach($utilisateurs as $utilisateur)
                                <tr>
                                    <td>{{ $utilisateur->cin }}</td><td>{{
$utilisateur->prenomNom }}</td>
                                    <td>{{ $utilisateur->dteNaissance }}</td><td>{{
$utilisateur->genre }}</td>
                                    <td>{{ $utilisateur->pays }}</td><td>{{
$utilisateur->email }}</td>
                                    <td>
                                    <button wire:click="edit({{ $utilisateur->id
}})" class="btn btn-primary btn-sm"
                                        data-bs-toggle="modal" data-bs-
target="#updateModal">Modifier</button>
                                    <button wire:click="delete({{ $utilisateur->id
}})" class="btn btn-danger btn-sm">Supprimer</button>
                                    </td>
                                </tr>
                                @endforeach

                            </tbody>
                        </table>
                        <div class="d-flex pb-0 pt-2 border  justify-content-
center">
                            {{ $utilisateurs->links() }}
                        </div>
                    </div>
                </div>
            </div>
        </div>

</div>
```

12. **resources/views/livewire/home.blade.php**

```
<!DOCTYPE html>
<html>
```

apcpedagogie.com

```
<head>
    <title></title>
    @livewireStyles
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css"
     rel="stylesheet" integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
      crossorigin="anonymous">
</head>
<body>

                        @if (session()->has('message'))
                            <div class="alert alert-success">
                                {{ session('message') }}
                            </div>
                        @endif
                        @livewire('utilisateurs')

    @livewireScripts
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
crossorigin="anonymous"></script>
 </body>
</html>
```

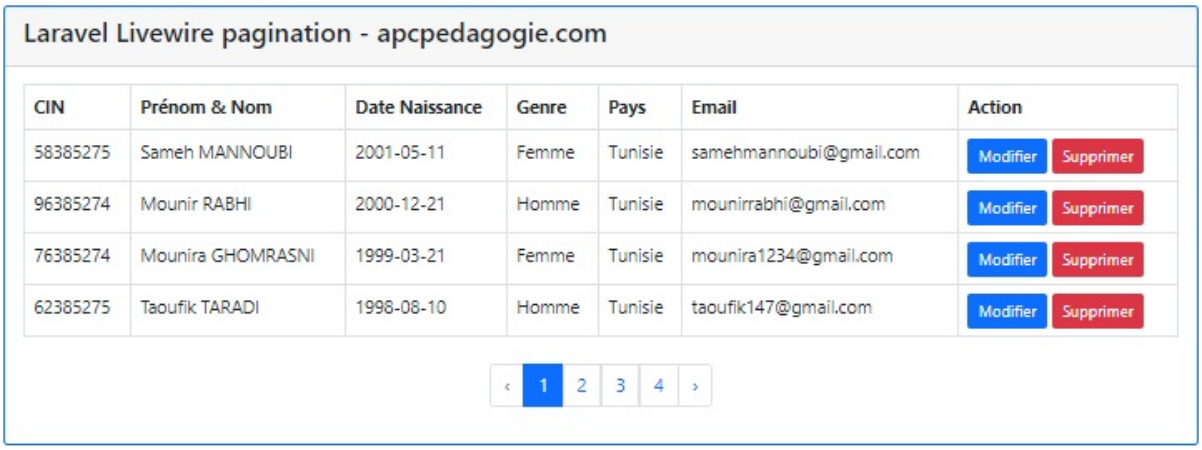
13. **Exécutez le serveur de développement Laravel 8**

    - Donc, tous sont des ensembles maintenant nous sommes prêts pour la sortie d'affichage dans le navigateur. Nous voulons donc d'abord démarrer le serveur Laravel. Nous devons donc aller à l'invite de commande et exécuter la commande suivante: `php artisan serve`

14. **Résultat**

    - Vous pouvez maintenant ouvrir l'URL ci-dessous sur votre navigateur: **`http://localhost:8000/utilisateurs`**
    - Le shéma suivant sera affiché

apcpedagogie.com



Sommaire du cours Laravel

Source : https://apcpedagogie.com/pagination-laravel-livewire · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com