

# Laravel Breeze

Guide de démarrage complet  
avec enregistrement Client

---

*Laravel 10 • Blade • MySQL • Tailwind CSS*

*Ce guide explique comment créer un projet Laravel Breeze et modifier le système d'inscription pour enregistrer un Client lié à chaque utilisateur.*

## 1. Mise en place du projet

### Étape 1 — Créer le projet Laravel 10

```
composer create-project laravel/laravel breezeProject "10.*"

cd breezeProject
```

*Télécharge Laravel 10 et crée le dossier breezeProject.*

### Étape 2 — Installer Laravel Breeze

```
composer require laravel/breeze --dev

php artisan breeze:install
```

*Installe l'authentification complète (login, register, reset password) avec Blade et Tailwind CSS.*

### Étape 3 — Installer les dépendances frontend

```
npm install

npm run dev
```

*Compile le CSS Tailwind et le JavaScript.*

### Étape 4 — Configurer la base de données

```
# Dans le fichier .env :

DB_DATABASE=breeze_db

DB_USERNAME=root

DB_PASSWORD=

# Lancer les migrations :

php artisan migrate
```

*Créez d'abord la base breeze\_db dans phpMyAdmin, puis lancez les migrations.*

### Étape 5 — Lancer le projet

```
php artisan serve

# Ouvrir dans le navigateur :
```

```
http://127.0.0.1:8000
```

*Le serveur de développement démarre sur le port 8000.*

## 2. Migration — Table clients

---

### Créer la migration :

```
php artisan make:migration create_clients_table
```

### Contenu de la migration (database/migrations/...create\_clients\_table.php) :

```
Schema::disableForeignKeyConstraints();

Schema::create('clients', function (Blueprint $table) {

    $table->id();

    $table->string('nom');

    $table->string('prenom');

    $table->string('tel');

    $table->string('adresse');

    $table->foreignId('user_id')->constrained()->onDelete('cascade');

    $table->timestamps();

});
```

*user\_id est une clé étrangère liée à la table users. Si un utilisateur est supprimé, son Client l'est aussi automatiquement (cascade).*

### Lancer la migration :

```
php artisan migrate
```

### 3. Modèle Client

---

#### Créer le modèle :

```
php artisan make:model Client
```

#### Contenu du fichier app/Models/Client.php :

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\User;

class Client extends Model
{
    use HasFactory;

    protected $fillable = [
        'nom', 'prenom', 'tel', 'adresse', 'user_id'
    ];

    public function user() {
        return $this->belongsTo(User::class);
    }
}
```

*\$fillable* liste les champs autorisés à l'écriture en masse. La relation *belongsTo* permet d'accéder au *User* depuis un *Client* : `$client->user->email`

## 4. Vue register.blade.php

Fichier : resources/views/auth/register.blade.php Remplacer le champ 'name' par les quatre champs du Client :

```
<!-- Nom -->

<x-input-label for="nom" :value="__('Nom')" />

<x-text-input id="nom" name="nom" type="text" :value="old('nom')" required />

<x-input-error :messages="$errors->get('nom')" />


<!-- Prénom -->

<x-input-label for="prenom" :value="__('Prenom')" />

<x-text-input id="prenom" name="prenom" type="text" :value="old('prenom')"
required />

<x-input-error :messages="$errors->get('prenom')" />


<!-- Téléphone -->

<x-input-label for="tel" :value="__('Telephone')" />

<x-text-input id="tel" name="tel" type="tel" :value="old('tel')" required />

<x-input-error :messages="$errors->get('tel')" />


<!-- Adresse -->

<x-input-label for="adresse" :value="__('Adresse')" />

<x-text-input id="adresse" name="adresse" type="text" :value="old('adresse')"
required />

<x-input-error :messages="$errors->get('adresse')" />


<!-- Email, Password, Confirm Password (inchangés) -->
```

*old('champ') permet de re-remplir le formulaire si la validation échoue.*

## 5. Contrôleur — RegisteredUserController.php

Fichier : app/Http/Controllers/Auth/RegisteredUserController.php

**Ce qui change par rapport au contrôleur Breeze par défaut :**

### 1. Ajouter l'import du modèle Client

```
use App\Models\Client;
```

### 2. Nouvelles règles de validation

```
'nom' => ['required', 'string', 'max:255'],

'prenom' => ['required', 'string', 'max:255'],

'tel' => ['required', 'string', 'max:20'],

'adresse' => ['required', 'string', 'max:255'],
```

### 3. Création du User (le champ name = nom + prenom)

```
$user = User::create([

    'name' => $request->nom . ' ' . $request->prenom,

    'email' => $request->email,

    'password' => Hash::make($request->password),

]);
```

### 4. Création du Client lié au User

```
Client::create([

    'nom' => $request->nom,

    'prenom' => $request->prenom,

    'tel' => $request->tel,

    'adresse' => $request->adresse,

    'user_id' => $user->id,

]);
```

## 6. Contrôleur complet (prêt à copier)

---

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use App\Models\Client;

use App\Models\User;

use App\Providers\RouteServiceProvider;

use Illuminate\Auth\Events\Registered;

use Illuminate\Http\RedirectResponse;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

use Illuminate\Validation\Rules;

use Illuminate\View\View;

class RegisteredUserController extends Controller
{
    public function create(): View
    {
        return view('auth.register');
    }
}
```

```

public function store(Request $request): RedirectResponse
{
    $request->validate([

        'nom' => ['required', 'string', 'max:255'],

        'prenom' => ['required', 'string', 'max:255'],

        'tel' => ['required', 'string', 'max:20'],

        'adresse' => ['required', 'string', 'max:255'],

        'email' => ['required', 'string', 'lowercase', 'email',

            'max:255', 'unique:'.User::class],

        'password' => ['required', 'confirmed', Rules\Password::defaults()],

    ]);

    $user = User::create([

        'name' => $request->nom.' '.$request->prenom,

        'email' => $request->email,

        'password' => Hash::make($request->password),

    ]);

    Client::create([

        'nom' => $request->nom,

        'prenom' => $request->prenom,

        'tel' => $request->tel,

        'adresse' => $request->adresse,

```

```

        'user_id' => $user->id,

    ]);

    event(new Registered($user));

    Auth::login($user);

    return redirect(RouteServiceProvider::HOME);
}
}

```

## Résumé du flux d'inscription

---

```

Formulaire --> Validation --> User::create() --> Client::create()

|

Auth::login($user) <-----+

|

Redirection /dashboard

```