

# Lab : Création d'une Application avec Validation de Formulaire

## Objectif

Créer une application web où un utilisateur peut soumettre un formulaire contenant son nom, son âge et son email. L'application validera les données soumises avant d'afficher un message de confirmation.

## Étapes du La

### Étape 1 : Installation de Laravel

1. Installez Laravel via Composer si ce n'est pas déjà fait :

```
composer create-project laravel/laravel laravel-form-validation-app
cd laravel-form-validation-app
```

2. Lancez le serveur de développement :

```
php artisan serve
```

Accédez à <http://localhost:8000> pour vérifier que l'application fonctionne.

### Étape 2 : Création des Routes

Ouvrez le fichier `routes/web.php` et ajoutez les routes nécessaires :

php

```
use App\Http\Controllers\FormController;

// Route pour afficher le formulaire
Route::get('/form', [FormController::class, 'showForm'])->
    >name('form.show');

// Route pour traiter la soumission du formulaire
Route::post('/form/submit', [FormController::class,
    'handleForm'])->name('form.submit');
```

### Étape 3 : Création du Contrôleur

1. Créez un contrôleur nommé `FormController` :

```
php artisan make:controller FormController
```

2. Ouvrez le fichier `app/Http/Controllers/FormController.php` et ajoutez les méthodes suivantes :  
php

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;

class FormController extends Controller
{
    // Afficher le formulaire
    public function showForm()
    {
        return view('form');
    }

    // Traiter la soumission du formulaire
    public function handleForm(Request $request)
    {
        // Valider les données du formulaire
        $validated = $request->validate([
            'name' => 'required|string|max:255',
            'age' => 'required|integer|min:1|max:120',
            'email' => 'required|email|max:255',
        ]);

        // Si la validation réussit, rediriger vers une page
        de confirmation
        return view('success', [
            'name' => $validated['name'],
            'age' => $validated['age'],
            'email' => $validated['email'],
        ]);
    }
}
```

#### Étape 4 : Création des Vues

1. **Vue du Formulaire** (`resources/views/form.blade.php`)  
Créez une nouvelle vue pour le formulaire :  
html

```
<!DOCTYPE html>
<html>
<head>
    <title>Formulaire</title>
</head>
<body>
```

```

<h1>Remplissez le formulaire</h1>

<!-- Afficher les messages d'erreur -->
@if ($errors->any())
    <div style="color: red;">
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif

<form action="{{ route('form.submit') }}" method="POST">
    @csrf <!-- Token CSRF pour la sécurité -->

    <label for="name">Nom :</label>
    <input type="text" name="name" id="name" value="{{
old('name') }}" required><br><br>

    <label for="age">Âge :</label>
    <input type="number" name="age" id="age" value="{{
old('age') }}" required><br><br>

    <label for="email">Email :</label>
    <input type="email" name="email" id="email" value="{{
old('email') }}" required><br><br>

    <button type="submit">Soumettre</button>
</form>
</body>
</html>

```

## 2. Vue de Confirmation (resources/views/success.blade.php)

Créez une vue pour afficher les résultats du formulaire :  
html

```

<!DOCTYPE html>
<html>
<head>
    <title>Résultats</title>
</head>
<body>
    <h1>Merci pour votre soumission !</h1>
    <p>Nom : {{ $name }}</p>
    <p>Âge : {{ $age }}</p>
    <p>Email : {{ $email }}</p>

    <a href="{{ route('form.show') }}">Retour au
formulaire</a>

```

```
</body>
</html>
```

## Étape 5 : Tester l'Application

1. Accédez à `http://localhost:8000/form` dans votre navigateur.
2. Remplissez le formulaire avec des données valides (nom, âge et email) et soumettez-le.
3. Vous serez redirigé vers une page de confirmation affichant les données soumises.
4. Essayez de soumettre le formulaire avec des données invalides (par exemple, un âge négatif ou un email incorrect). Vous verrez des messages d'erreur correspondants.

## Explications des Composants

1. **Routes :**
  - Les routes définissent les points d'entrée de l'application et associent chaque URL à une méthode spécifique du contrôleur.
2. **Contrôleur :**
  - Le contrôleur gère les requêtes entrantes, valide les données du formulaire et renvoie les vues appropriées.
  - La méthode `handleForm` utilise la méthode `validate` de Laravel pour valider les données du formulaire.
3. **Vues :**
  - `form.blade.php` : Affiche le formulaire et affiche les messages d'erreur en cas de validation échouée.
  - `success.blade.php` : Affiche les données validées après soumission réussie.
4. **Validation de Formulaire :**
  - Laravel fournit une syntaxe concise pour valider les données du formulaire dans le contrôleur.
  - Les règles de validation sont définies dans le tableau passé à la méthode `validate`.

## Règles de Validation Utilisées

CHAMP	REGLE(S)	DESCRIPTION
<code>name</code>	<code>`required`</code>	<code>string</code>
<code>age</code>	<code>`required`</code>	<code>integer</code>
<code>email</code>	<code>`required`</code>	<code>email</code>

## Améliorations Possibles

1. **Personnalisation des Messages d'Erreur :** Vous pouvez personnaliser les messages d'erreur en passant un tableau supplémentaire à la méthode `validate` :  
`php`

```
$validated = $request->validate([
    'name' => 'required|string|max:255',
    'age' => 'required|integer|min:1|max:120',
    'email' => 'required|email|max:255',
], [
    'name.required' => 'Le champ Nom est requis.',
    'age.min' => 'L\'âge doit être supérieur à 0.',
    'email.email' => 'Veuillez entrer une adresse email valide.',
]);
```

2. **Ajout de Styles CSS** : Ajoutez des styles CSS personnalisés pour rendre l'interface utilisateur plus attrayante.
3. **Utilisation de Blade Directives** : Utilisez des directives Blade comme `@error` pour afficher les messages d'erreur de manière plus propre :  
html

```
@error('name')
    <div style="color: red;">{{ $message }}</div>
@enderror
```

## Code Final

### Contrôleur ( `FormController.php` )

php

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;

class FormController extends Controller
{
    public function showForm()
    {
        return view('form');
    }

    public function handleForm(Request $request)
    {
        $validated = $request->validate([
            'name' => 'required|string|max:255',
            'age' => 'required|integer|min:1|max:120',
            'email' => 'required|email|max:255',
        ]);

        return view('success', [
```

```

        'name' => $validated['name'],
        'age' => $validated['age'],
        'email' => $validated['email'],
    ]);
}
}

```

## Vue du Formulaire ( `form.blade.php` )

html

```

<!DOCTYPE html>
<html>
<head>
    <title>Formulaire</title>
</head>
<body>
    <h1>Remplissez le formulaire</h1>

    <!-- Afficher les messages d'erreur -->
    @if ($errors->any())
        <div style="color: red;">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form action="{{ route('form.submit') }}" method="POST">
        @csrf

        <label for="name">Nom :</label>
        <input type="text" name="name" id="name" value="{{
old('name') }}" required><br><br>

        <label for="age">Âge :</label>
        <input type="number" name="age" id="age" value="{{
old('age') }}" required><br><br>

        <label for="email">Email :</label>
        <input type="email" name="email" id="email" value="{{
old('email') }}" required><br><br>

        <button type="submit">Soumettre</button>
    </form>
</body>
</html>

```

## Vue de Confirmation ( `success.blade.php` )

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Résultats</title>
</head>
<body>
    <h1>Merci pour votre soumission !</h1>
    <p>Nom : {{ $name }}</p>
    <p>Âge : {{ $age }}</p>
    <p>Email : {{ $email }}</p>

    <a href="{{ route('form.show') }}">Retour au
formulaire</a>
</body>
</html>
```

## Conclusion

Ce laboratoire vous a permis de créer une application Laravel utilisant les concepts suivants :

- **Vues** : Pour afficher les formulaires et les résultats.
- **Formulaires** : Pour collecter les données de l'utilisateur.
- **Routes** : Pour mapper les URLs aux actions du contrôleur.
- **Contrôleurs** : Pour traiter les requêtes et interagir avec les vues.
- **Validation de Formulaire** : Pour garantir que les données soumises respectent les critères définis.