# Validation de formulaire avec Laravel

[sommaire]

1. **Objectifs**

- Connaitre les méthodes de validation d'un formulaire avec Laravel.
- Etre capable de valider un formulaire laravel.

- **Présentation**

- La validation de formulaire est un fait très important pour assainir et protéger les données indésirables dans notre application.
- Laravel propose plusieurs approches différentes pour valider les données entrantes de votre application. Par défaut, la classe de contrôleur de base de Laravel utilise un trait `ValidatesRequests` qui fournit une méthode pratique pour valider les requêtes HTTP entrantes avec une variété de règles de validation puissantes.
- Dans notre tutoriel précédent, nous avons mis en place un formulaire assez sympa pour nous permettre de soumettre un nouveau inscrit à notre base de données d'inscription. Dans cet article, je vais vous montrer les différentes méthodes de validation Laravel pour valider la demande de formulaire utilisateur.
- Il existe plusieurs méthodes qui permet la validation de formulaire via Laravel.

1. Validation manuelle
2. Validation du contrôleur
3. Demande de validation
4. Validation du modèle

- **Validation manuelle**

1. **Présentation**

- Il s'agit d'un processus de validation manuelle avec la classe de façade de validation Laravel.
- Le moyen le plus simple de valider la validation du formulaire consiste à utiliser la façade `Validator` .
- Vous pouvez définir vos règles de validation dans votre méthode avec la façade Validateur.

apcpedagogie.com

- Si vous ne souhaitez pas utiliser la méthode `ValidatesRequests` du class `validate`, vous pouvez créer une instance de validateur manuellement en utilisant la **Validator** façade .

- **Exemple**

  - **Routes**
  - On va donc avoir besoin de 2 routes :

```
Route::get('/form-inscription', 'App\Http\Controllers\EtudiantController@createUserForm');
Route::post('/form-inscription', 'App\Http\Controllers\EtudiantController@store');
```

- **Le contrôleur**

```
<?php

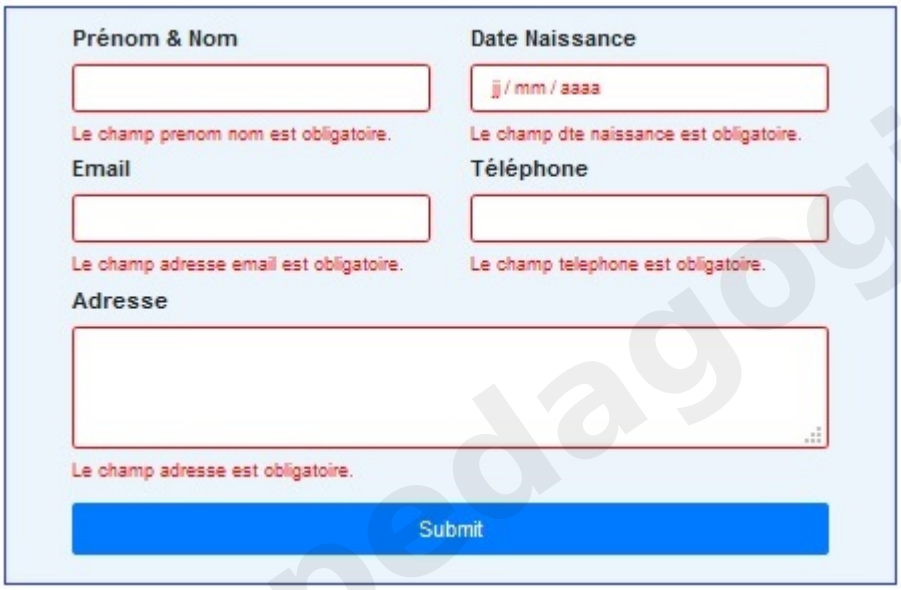
namespace App\Http\Controllers;
use Validator;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
//use App\Etudiant;
class EtudiantController extends Controller
{
    // Create Form
  public function createUserForm(Request $request) {
    return view('form-inscription');
  }
  // Store Form data in database
  public function store(Request $request) {
      // Form validation
      $validator = Validator::make($request->all(), [
          'prenomNom' => 'required',
          'dteNaissance'=>'required',
          'email' => 'required|email',
          'telephone' => 'required|regex:/^([0-9\s\-\+\(\)]*)$/|min:8',
          'adresse' => 'required'
       ]);
      //  Store data in database
      //'App\Models\Etudiant'::create($request->all());
      //
      //return back()->with('success', 'Votre formulaire a été soumis.');
if($validator->fails()){
    return redirect('form-inscription')->withErrors($validator)->withInput();
```

apcpedagogie.com

```
}
}
    }
```

- Le code ci-dessus passera par les règles définies pour la demande particulière et redirigera vers la page si la demande a échoué. Vous pouvez afficher les erreurs en ajoutant les codes suivants à votre modèle de vue.

```
            @if ($errors->has('dteNaissance'))
                <div class="error">
                    {{ $errors->first('dteNaissance') }}
                </div>
            @endif
```




## Validation avec des contrôleurs

1. **Présentation**

- Dans cette méthode, nous pouvons valider notre demande de formulaire soumise par l'utilisateur directement dans notre contrôleur méthode par la méthode `$this->validate()`.
- Dans cette méthode, si la validation n'est pas réussie, elle sera automatiquement redirigée avec toutes les erreurs de validation.

- **Exemple**

apcpedagogie.com

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
//use App\Etudiant;
class EtudiantController extends Controller
{
    // Create Form
  public function createUserForm(Request $request) {
    return view('form-inscription');
  }
  // Store Form data in database
  public function UserForm(Request $request) {

      // Form validation
      $this->validate($request, [
          'prenomNom' => 'required',
          'dteNaissance'=>'required',
          'email' => 'required|email',
          'telephone' => 'required|regex:/^([0-9\s\-\+\(\)]*)$/|min:8',
          'adresse' => 'required'
       ]);

      //  Store data in database
      'App\Models\Etudiant'::create($request->all());
      //
      return back()->with('success', 'Votre formulaire a été soumis.');
  }
}
```

- **Activité**
  - Créez un fichier de vue dans le répertoire '**resources/views**' avec le nom '**loginValidate.blade.php**' et écrivez le code ci-dessous dans ce fichier.

```
<form method="post" action="/loginValidate">
        @csrf
        <div class="row mt-1">
        <div class="form-group col-6 lg">
        <input type="text" name="username"
```

apcpedagogie.com

```
                placeholder="Entrer votre login" class="form-control">
        </div>
        <div class="form-group col-6 lg">
        <input type="password" name="password"
                placeholder="Entrer votre mot de passe" class="form-control">
        </div>
        </div>
        <div class="row mt-1">
            <div class="form-group col-12 lg">
        <input type="submit" name="submit" value="Submit" class="btn btn-primary btn-block">
    </div>
    </div>
    </form>
    <ul class="error">
        @foreach($errors->all() as $error)
        <li>{{ $error }}</li>
        @endforeach
    </ul>
    </div>
```

- Créez un contrôleur dans le répertoire '**app/Http/Controllers**' avec le nom '**LoginValidateController.php**'.
- Utilisez la commande '`php artisan make:controller LoginValidateController`' pour créer ce contrôleur.
- Écrivez le code de validation ci-dessous pour valider les données soumises par le formulaire.

```
class LoginValidateController extends Controller
{
    //
    public function getValidate() {
        return view('loginValidate');
    }

    public function postValidate(Request $request) {
        $request->validate([
            'username' => 'required',
            'password' => 'required|min:8|max:255',
        ]);
    }
}
```

- Vous remarquze certainement, "required" est utilisée pour donner à l'utilisateur une erreur lorsqu'il laisse le champ vide. Et 'min' correspond au minimum de caractères et 'max' correspond au maximum de caractères pouvant être saisis.

apcpedagogie.com

- Écrivez le code ci-dessous dans le fichier '`web.php`' du répertoire '**routes**'

```
Route :: get ('/loginValidate', 'App\Http\Controllers\LoginValidateController@getValidate');
Route :: post ('/loginValidate', 'App\Http\Controllers\LoginValidateController@postValidate');
```

- Ici, nous avons spécifié la méthode '`get`' pour afficher la vue de notre page de connexion et la méthode '`post`' pour la validation des données d'entrée soumises par l'utilisateur via le formulaire HTML.

---

**Sommaire du cours Laravel**

Source : https://apcpedagogie.com/validation-de-formulaire-avec-laravel · téléchargé le 04/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com