

# S3 : Introduction à la validation des données en Laravel

Ressources :

<https://laravel.com/docs/11.x/validation>

## Introduction à la validation des données en Laravel

- La validation des données est essentielle pour s'assurer que les utilisateurs soumettent des informations correctes et sûres dans une application.
- Laravel fournit des moyens pour valider les données des utilisateurs.

## Validation côté serveur avec Laravel :

- La méthode `validate()` :
  - Méthode intégrée pour valider les données.
  - Elle prend en paramètre un tableau de règles de validation et un tableau de données (souvent les données d'un formulaire).

```
$request->validate([  
    'name' => 'required|max:255',  
    'model' => 'required|min:3',  
]);
```

- **Règles courantes de validation :**
  - `required` : le champ est obligatoire.
  - `max:X` : la longueur maximale du champ (X étant un nombre).
  - `min:X` : la longueur minimale du champ (X étant un nombre).
  - `email` : le champ doit être une adresse email valide.
  - `unique:table,column` : le champ doit être unique dans la table spécifiée.

# Sécurisation des applications Laravel

## 1. Validation côté serveur :

- La validation des données côté serveur est importante, même si les données sont validées côté client. Elle empêche des données malveillantes ou incorrectes d'entrer dans la base de données.

## 2. Protection des routes :

- **Middleware** : Laravel propose des middlewares pour protéger des routes ou des actions contre des utilisateurs non autorisés, comme le middleware `auth` qui protège les pages réservées aux utilisateurs connectés.

Exemple :

```
Route::middleware(['auth'])->group(function () {  
    Route::get('/motorcycle/create', [MotorcycleController::class, 'create']);  
});
```

## 3. Exercices (20 min)

### 1. Ajouter une validation de formulaire pour la création d'une nouvelle moto

**Objectif** : Ajouter des règles de validation pour un formulaire de création de moto.

#### Instructions :

1. Créez un formulaire HTML pour la création d'une nouvelle moto avec les champs suivants :
  - **Nom** (required, max: 255 caractères)
  - **Modèle** (required, min: 3 caractères)
2. Dans le contrôleur correspondant, appliquez les règles de validation sur ces champs avant de sauvegarder les données dans la base de données.

```
$request->validate([
```

```
'name' => 'required|max:255',  
'model' => 'required|min:3',  
]);
```

## 2. Sécuriser les routes

**Objectif :** Ajouter une protection sur les routes d'ajout de moto pour s'assurer qu'elles ne sont accessibles qu'aux utilisateurs connectés.

**Instructions :**

1. Dans votre `MotorcycleController`, assurez-vous que la route de création de moto est protégée par le middleware `auth`.

Exemple :

```
Route::middleware(['auth'])->group(function () {  
    Route::get('/motorcycle/create', [MotorcycleController::class, 'create']);  
    Route::post('/motorcycle', [MotorcycleController::class, 'store']);  
});
```

2. Assurez-vous que seules les personnes connectées puissent créer une moto.

## 4. Projet Fil Rouge (30 min)

**Objectif :** Ajouter un formulaire de création de moto dans l'application

1. **Créer une route pour la page de création d'une moto :**

- Dans `routes/web.php`, ajoutez une route pour afficher le formulaire de création de moto.

```
Route::get('/motorcycle/create', [MotorcycleController::class, 'create']);
```

2. **Créer un formulaire dans la vue `create.blade.php` :**

- Créez un formulaire HTML avec les champs `name` et `model`. Assurez-vous que la validation des données est effectuée avant de soumettre le formulaire.

### 3. Créer la méthode `store()` dans le contrôleur :

- La méthode `store()` doit valider les données et les enregistrer dans la base de données.

```
public function store(Request $request)
{
    $request->validate([
        'name' => 'required|max:255',
        'model' => 'required|min:3',
    ]);

    Motorcycle::create($request->all());

    return redirect()->route('motorcycle.index');
}
```

### 4. Afficher les erreurs de validation :

- Si des erreurs de validation se produisent, elles doivent être affichées au-dessus du formulaire dans la vue `create.blade.php`.

```
@if ($errors->any())
<div>
<ul>
    @foreach ($errors->all() as $error)
        <li>{{ $error }}</li>
    @endforeach
</ul>
</div>
@endif
```

