

Procédés de Validation dans Laravel

Dans Laravel, lorsque vous définissez des règles de validation dans un modèle, vous avez deux options principales : vous pouvez utiliser les règles de validation directement dans un **Form Request** ou les intégrer au niveau de votre modèle. Habituellement, la validation se fait dans les **Form Requests**, mais pour centraliser ou réutiliser certaines règles, vous pouvez créer une méthode dans votre modèle pour les gérer. Voici comment vous pouvez le faire dans les deux cas.

Option 1 : Utilisation de Form Request (recommandée)

Cette méthode consiste à créer une classe de Form Request pour valider les données avant qu'elles ne soient transmises à votre modèle.

```
/** @test */
public function store_validates_using_a_form_request()
{
    $this->assertActionUsesFormRequest(
        NewsletterSubscriptionController::class,
        'store',
        StoreNewsletterSubscription::class
    );
}
```

Étapes :

1. Créez un Form Request à l'aide de la commande artisan :
`php artisan make:request StoreUserRequest`
2. Modifiez le fichier généré dans `app/Http/Requests/StoreUserRequest.php` pour ajouter vos règles de validation :

```
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class StoreUserRequest extends FormRequest
{
```

```

public function authorize()
{
    // Assurez-vous que l'utilisateur est autorisé à faire
    // cette requête.
    return true;
}

public function rules()
{
    return [
        'name' => 'required|string|max:255',
        'email' => 'required|email|unique:users,email',
        'password' => 'required|string|min:8|confirmed',
    ];
}
}

```

3. Dans votre contrôleur, vous pouvez utiliser ce Form Request :

```

use App\Http\Requests\StoreUserRequest;

class UserController extends Controller
{
    public function store(StoreUserRequest $request)
    {
        // Les données seront validées avant cette étape
        $data = $request->validated();

        // Création de l'utilisateur
        User::create($data);

        // Redirection ou autre traitement
    }
}

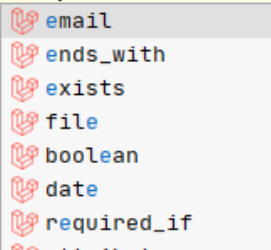
```

Option 2 : Validation dans le Modèle

Bien que Laravel ne favorise pas directement la validation dans les modèles, vous pouvez centraliser certaines règles en créant une méthode dans votre

modèle. Voici un exemple :

```
public function rules()
{
    return [
        'email' => ['required', 'email'],
        'password' => 'required|password',
        'name' => 'required|string|max:255',
        'other_field' => [
            'required_if:name',
        ],
    ];
}
```



Étapes :

1. Ajoutez une méthode `rules()` dans votre modèle :

```
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    protected $fillable = ['name', 'email', 'password'];

    public static function rules()
    {
        return [
            'name' => 'required|string|max:255',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|string|min:8|confirmed',
        ];
    }
}
```

2. Vous pouvez utiliser ces règles dans un contrôleur ou un service lors de la validation des données :

```
use Validator;
```

```

use App\Models\User;

class UserController extends Controller
{
    public function store(Request $request)
    {
        // Valider avec les règles définies dans le modèle
        $validator = Validator::make($request->all(), User::rules());

        if ($validator->fails()) {
            return redirect()->back()->withErrors($validator)->withInput();
        }

        // Les données sont valides, on crée l'utilisateur
        User::create($request->all());
    }
}

```

Conclusion

Bien que la validation dans les modèles soit possible, Laravel encourage l'utilisation des **Form Requests** pour séparer proprement la logique de validation de votre logique métier. Cela permet une meilleure organisation de votre code, et vous permet de centraliser toute la validation dans des classes dédiées.