

# Form Validation in Laravel

## 1. Validation via Request Class

- **Create a Form Request Class:**

```
php artisan make:request ContactFormRequest
```

- **Define Validation Rules and Messages:**

```
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class ContactFormRequest extends FormRequest
{
    public function authorize()
    {
        return true; // Assuming no specific authorization check is needed
    }

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

```
return [ 'name' => 'required|string|max:255', 'email' => 'required|email|max:255', 'subject' =>
'required|string|max:255', 'message' => 'required|string', ];
public function messages() { return [ 'name.required' => 'The name field is required.']; }
```

- **Use the Form Request in Your Controller:**

```
use App\Http\Requests\ContactFormRequest;

class ContactController extends Controller
{
    public function submit(ContactFormRequest $request)
    {
        // The request will be validated automatically

        // Retrieve the validated input data...
        $validated = $request->validated();

        // Perform the desired operations, such as sending an email or saving to the database
    }
}
```

```
$validated = $request->validated();
```

## 2. Inline Validation within Controller

```
public function submit(Request $request)
{
    $validator = Validator::make($request->all(), [
        'name' => 'required|string|max:255',
        'email' => 'required|email|max:255',
    ], [
        'name.required' => 'The name field is required.',
        'name.string' => 'The name must be a string.',
        'name.max' => 'The name may not be greater than 255 characters.',
        'email.required' => 'We need to know your e-mail address!',
        'email.email' => 'The email must be a valid email address.',
        'email.max' => 'The email may not be greater than 255 characters.',
    ]);

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

### 3 Handle Error in blade

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

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

```
@error('name')
    <div class="alert alert-danger">{{ $message }}</div>
@enderror
```

```
@error('name') <div class="alert alert-danger">{{ $message }}</div> @enderror
```

```
{{route('savecontactus')}}}
```

```
php artisan make:request ContactFormRequest
```

Here is a list of basic validation rule names in Laravel:

1. `accepted`
2. `active_url`
3. `after:date`

4. `after_or_equal:date`
5. `alpha`
6. `alpha_dash`
7. `alpha_num`
8. `array`
9. `before:date`
10. `before_or_equal:date`
11. `between:min,max`
12. `boolean`
13. `confirmed`
14. `date`
15. `date_equals:date`
16. `date_format:format`
17. `different:field`
18. `digits:value`
19. `digits_between:min,max`
20. `dimensions`
21. `distinct`
22. `email`
23. `exists:table,column`
24. `file`
25. `filled`
26. `image`
27. `in:foo,bar,...`
28. `in_array:anotherfield.*`
29. `integer`
30. `ip`
31. `ipv4`
32. `ipv6`
33. `json`
34. `max:value`
35. `mimes:foo,bar,...`
36. `mimetypes:foo,bar,...`
37. `min:value`
38. `not_in:foo,bar,...`
39. `nullable`
40. `numeric`
41. `present`

- 42. `regex:pattern`
- 43. `required`
- 44. `required_if:anotherfield,value`
- 45. `required_unless:anotherfield,value`
- 46. `required_with:foo,bar,...`
- 47. `required_with_all:foo,bar,...`
- 48. `required_without:foo,bar,...`
- 49. `required_without_all:foo,bar,...`
- 50. `same:field`
- 51. `size:value`
- 52. `string`
- 53. `timezone`
- 54. `unique:table,column`
- 55. `url`
- 56. `uuid`