

Laravel - Forms

Topics: Laravel

Written on December 25, 2023

In Laravel, working with forms is made easy with the help of the Blade templating engine and Laravel Collective's HTML and Form Builders. Here's a guide on working with forms in Laravel:

1. Creating Forms:

Forms can be created using the form method provided by Laravel Collective. Laravel Collective provides a set of HTML and Form helper methods that simplify the process of creating and working with HTML forms.

First, you need to install Laravel Collective:

```
composer require laravelcollective/html
```

After installation, you should add the service provider and facade in the config/app.php file:

```
'providers' => [
    // ...
    Collective\Html\HtmlServiceProvider::class,
],
'aliases' => [
    // ...
    'Form' => Collective\Html\FormFacade::class,
    'Html' => Collective\Html\HtmlFacade::class,
],
```

2. Creating a Form:

In your Blade view, you can use the Form facade to create forms.

```
{!! Form::open(['url' => '/submit', 'method' => 'post']) !!}
   // Form fields and controls go here
{!! Form::close() !!}
```

3. Form Fields:

Laravel Collective provides methods for various form fields:

```
{!! Form::text('name', null, ['class' => 'form-control']) !!}
{!! Form::password('password', ['class' => 'form-control']) !!}
{!! Form::email('email', null, ['class' => 'form-control']) !!}
{!! Form::select('role', ['admin' => 'Admin', 'user' => 'User'], null,
['class' => 'form-control']) !!}
```

4. CSRF Protection:

Laravel automatically adds CSRF protection to your forms. Use the @csrf directive to include the CSRF token.

```
{!! Form::open(['url' => '/submit', 'method' => 'post']) !!}
  @csrf
  // Form fields and controls go here
{!! Form::close() !!}
```

5. Form Validation:

You can validate form data using Laravel's built-in validation features. The validate method can be called in your controller to validate the incoming request.

```
public function store(Request $request)
{
    $request->validate([
          'name' => 'required|string|max:255',
          'email' => 'required|email|unique:users',
          'password' => 'required|min:8',
    ]);

// Process the form data
}
```

6. Old Input:

After a form submission with validation errors, you can repopulate the form fields with old input using the old method.

```
{!! Form::text('name', old('name'), ['class' => 'form-control']) !!}
```

7. Form Actions:

Set the action attribute of the form to the URL where the form data should be submitted.

```
{!! Form::open(['url' => '/submit', 'method' => 'post']) !!}
   // Form fields and controls go here
   {!! Form::submit('Submit', ['class' => 'btn btn-primary']) !!}
{!! Form::close() !!}
```

```
💏 app.php
LARAVEL
> bootstrap

✓ config

e app.php
 auth.php
                                                                 Illuminate\Auth\AuthServiceProvider::class,
 m broadcasting.php
                                                                 {\tt Illuminate} \\ {\tt Broadcasting} \\ {\tt BroadcastServiceProvider::class},
💏 cache.php
                                                                 Illuminate\Bus\BusServiceProvider::class,
                                                                 er cors.php
                                                                 Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
 ndatabase.php
                                                                 Illuminate\Cookie\CookieServiceProvider::class,
😭 filesystems.php
                                                                 Illuminate\Database\DatabaseServiceProvider::class,
nashing.php
                                                                 Illuminate\Encryption\EncryptionServiceProvider::class,
 R logging.php
                                                                 Illuminate\Filesystem\FilesystemServiceProvider::class,
💏 mail.php
                                                                 Illuminate\Foundation\Providers\FoundationServiceProvider::class,
🗬 queue.php
                                                                 Illuminate \verb|\Hashing\HashServiceProvider::class|,
                                                                 Illuminate\Mail\MailServiceProvider::class,
🐄 sanctum.php
                                                                 Illuminate\Notifications\NotificationServiceProvider::class,
services.php
                                                                 Illuminate\Pagination\PaginationServiceProvider::class,
💏 session.php
                                                                 Illuminate\Pipeline\PipelineServiceProvider::class,
💏 view.php
                                                                 {\tt Illuminate \backslash Queue \backslash Queue Service Provider:: class,}
> database
                                                                 Illuminate \verb|\Redis \verb|\Redis Service Provider::class|,
                                                                 Illuminate\Auth\Passwords\PasswordResetServiceProvider::class,
> lang
                                                                 Illuminate\Session\SessionServiceProvider::class,
                                                                 Illuminate\Translation\TranslationServiceProvider::class,
> resources
                                                                 Illuminate\Validation\ValidationServiceProvider::class,
> routes
                                                                 Illuminate\View\ViewServiceProvider::class,
> storage
> tests
OUTLINE
```

© Copyright Aryatechno. All Rights Reserved. Written tutorials and materials by Aryatechno