

Laravel Filament Admin Panel Structure

Admin panel with Filament. Rapid CRUD, forms, tables, and dashboards.

#laravel #php #filament #admin #crud #backend

PNG

PDF

Copy

</> Prompt

Project Directory

myproject/

- artisan
- composer.json
- package.json
- vite.config.js
- .env
- .env.example
- app/
- Filament/
- Resources/
- UserResource/
- UserResource.php
- Pages/
- ListUsers.php
- CreateUser.php
- EditUser.php
- ProductResource/
- OrderResource/
- Pages/
- Dashboard.php
- Widgets/
- StatsOverview.php
- OrdersChart.php
- Http/
- Controllers/
- Middleware/
- Models/
- User.php
- Product.php
- Order.php
- Providers/
- Filament/
- bootstrap/
- config/
- database/
- resources/
- views/
- css/
- js/
- routes/
- tests/
- public/
- storage/

Why This Structure?

Filament generates beautiful admin panels from your Eloquent models. Define form fields and table columns in PHP, get a full CRUD interface. Includes widgets, charts, and custom pages for dashboards.

Key Directories

- app/Filament/Resources/** - One folder per model with CRUD pages
- app/Filament/Widgets/** - Dashboard stats and charts
- app/Filament/Pages/** - Custom admin pages beyond CRUD
- app/Models/** - Eloquent models Filament builds on

Filament Resource

```
// app/Filament/Resources/UserResource.php
public static function form(Form $form): Form
{
    return $form->schema([
        TextInput::make('name')->required(),
        TextInput::make('email')->email()->required(),
    ]);
}

public static function table(Table $table): Table
{
    return $table->columns([
        TextColumn::make('name')->sortable(),
        TextColumn::make('email')->searchable(),
    ]);
}
```

Getting Started

- composer create-project laravel/laravel myproject
- composer require filament/filament
- php artisan filament:install --panels
- php artisan make:filament-user
- php artisan make:filament-resource User

When To Use This

- Need admin panel quickly
- Internal tools and back-office apps
- CRUD-heavy applications
- E-commerce admin dashboards
- Want consistent UI without design work

Trade-offs

- Customization limits** - Complex custom UIs may fight the framework
- Learning Filament** - Form/Table builder DSL takes time to learn
- Opinionated UI** - Stuck with Filament design system

Naming Conventions

- Resources** - ModelNameResource (UserResource, ProductResource)
- Widgets** - DescriptiveWidget (StatsOverview, OrdersChart)
- Pages** - ActionModel (ListUsers, CreateUser, EditUser)