

# Laravel Inertia.js Project Structure

Modern monolith with Inertia.js. Server-side routing with Vue or React pages.

#laravel #php #inertia #vue #react #spa #full-stack

PNG

PDF

Copy

</> Prompt

## Project Directory

myproject/

- artisan
- composer.json
- package.json
- vite.config.js
- tailwind.config.js
- tsconfig.json If using TypeSc...
- .env
- .env.example
- > app/
- > Http/
- > Controllers/
- Controller.php
- DashboardController.php
- UserController.php
- Auth/
- Requests/
- Middleware/
- Models/
- Services/
- Providers/
- bootstrap/
- config/
- database/
- > resources/
- > js/ Frontend app
- app.js Inertia app ent...
- bootstrap.js
- > Pages/ Inertia pages (...)
- Dashboard.vue
- > Users/
- Index.vue
- Create.vue
- Edit.vue
- > Auth/
- Login.vue
- Register.vue
- > Components/ Reusable Vue/Re...
- Button.vue
- Modal.vue
- Pagination.vue
- > Layouts/
- AppLayout.vue
- GuestLayout.vue
- > Composables/ Vue composables
- useForm.js
- css/
- > views/
- app.blade.php Single Blade te...
- > routes/
- web.php
- auth.php
- tests/
- public/
- storage/

## Why This Structure?

Inertia.js gives you SPA experience with server-side routing. Controllers return Inertia responses, pages are Vue/React components. No API needed —data flows directly from controller to frontend. Best of both worlds.

## Key Directories

- resources/js/Pages/** - Vue or React page components
- resources/js/Components/** - Reusable UI components
- resources/js/Layouts/** - Persistent layouts across pages
- resources/views/app.blade.php** - Single Blade entry point

## </> Inertia Flow

```
// app/Http/Controllers/UserController.php
public function index()
{
    return Inertia::render('Users/Index', [
        'users' => User::paginate(10)
    ]);
}

// resources/js/Pages/Users/Index.vue
defineProps({ users: Object })
```

## >\_ Getting Started

- `composer create-project laravel/laravel myproject`
- `composer require inertiajs/inertia-laravel`
- `npm install @inertiajs/vue3`
- Create `resources/views/app.blade.php` with @inertia
- Configure `resources/js/app.js` with createInertiaApp

## ☑ When To Use This

- Want SPA UX without building an API
- Teams comfortable with Vue or React
- Monolith architecture preference
- Rapid development with Laravel + modern frontend
- Apps needing SEO (SSR supported)

## ⚖ Trade-offs

- Coupled stack** - Frontend tightly coupled to Laravel controllers
- No mobile API** - Need separate API for native apps
- Learning curve** - Must learn both Laravel and Vue/React