

set up a Laravel + React + Inertia.js full-stack application

1. Create a Laravel Project

```
composer create-project laravel/laravel laravel-react-inertia
cd laravel-react-inertia
```

2. Install Inertia.js (Server-Side for Laravel)

```
composer require inertiajs/inertia-laravel
```

Publish the middleware:

```
php artisan inertia:middleware
```

✓ Register Inertia Middleware in Laravel 12

```
composer require inertiajs/inertia-laravel
php artisan inertia:middleware
```

+ Step 1: Register Middleware in bootstrap/app.php

Open bootstrap/app.php, and add your middleware in the web middleware group like this:

```
use App\Http\Middleware\HandleInertiaRequests;
use Illuminate\Foundation\Application;
use Illuminate\Foundation\Configuration\Exceptions;
use Illuminate\Foundation\Configuration\Middleware;

return Application::configure(basePath: dirname(__DIR__))
    ->withRouting(
        web: __DIR__.'/../routes/web.php',
        api: __DIR__.'/../routes/api.php',
    )
    ->withMiddleware(function (Middleware $middleware) {
```

```
// Register global middleware here if needed

// Add the Inertia middleware to the 'web' group
$middleware->group('web', [
    \Illuminate\Cookie\Middleware\EncryptCookies::class,
    \Illuminate\Http\Middleware\AddLinkHeadersForPreloadedAssets::class,
    \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
    \Illuminate\Session\Middleware\StartSession::class,
    \Illuminate\View\Middleware\ShareErrorsFromSession::class,
    HandleInertiaRequests::class,
]);
})
->withExceptions(function (Exceptions $exceptions) {
    //
})->create();
```

✅ Now HandleInertiaRequests will be used in all web routes.

⚠️ If You're Using Laravel Breeze

```
php artisan breeze:install react
```

It **automatically registers** everything for you—including the Inertia middleware—so you **don't need to do anything extra**.

But if you're setting things up manually or modifying middleware, bootstrap/app.php is where it's done now.

3. Install Laravel Breeze with React + Inertia

```
composer require laravel/breeze --dev
php artisan breeze:install react
npm install && npm run dev
php artisan migrate
```

This is set up:

- Inertia.js

- React
- Auth scaffolding
- TailwindCSS
- Proper file structure

4. Directory Structure After Breeze Install

```
resources/  
├── js/  
│   ├── App.jsx      // Bootstraps Inertia + React  
│   ├── Pages/       // Your React “views”  
│   └── components/   // Optional React components  
routes/  
└── web.php           // Laravel routes using Inertia::render()
```

5. Add a Custom Route and Page

routes/web.php

```
use Inertia\Inertia;  
use Illuminate\Support\Facades\Route;  
  
Route::get('/', function () {  
    return Inertia::render('Home', [  
        'message' => 'Hello from Laravel with React + Inertia!'  
    ]);  
});
```

resources/js/Pages/Home.jsx

```
import React from 'react';

export default function Home({ message }) {
  return (
    <div className="p-6 text-center">
      <h1 className="text-2xl font-bold">{message}</h1>
    </div>
  );
}
```

6. Run Your App

```
php artisan serve
npm run dev
```

CODE HELP

Input Request File Validation ->

```
$validate = Validator::make($request->all(),[
    'title' => 'required|string|max:255',
    'slug' => ['required', 'string', 'max:255'],
    "thumbnail"    => [
        'image',
        'mimes:jpg,png,jpeg,gif,svg,webp',
        'dimensions:min_width=550,min_height=350,max_width=550,max_height=350'
    ],
]);
if($validate->fails()){
    toastr()->error($validate->messages());
    return back();
}
```

Store Image ->

```
if ($request->file('thumbnail')){
    $thumbnail = $request->file('thumbnail');
    $thumbnailName = $thumbnail->getClientOriginalName();
    $directory = 'upload/article/';
    $thumbnail->move($directory,$thumbnailName);
    $thumbnailUrl = $directory.$thumbnailName;
    $article->thumbnail = $thumbnailUrl;
}
```

Update Image ->

```
if ($request->file('thumbnail')){
    if (file_exists($article->thumbnail)){
        unlink($article->thumbnail);
    }
    $thumbnail = $request->file('thumbnail');
    $thumbnailName = $thumbnail->getClientOriginalName();
    $directory = 'upload/product/thumbnail/';
    $thumbnail->move($directory,$thumbnailName);
    $thumbnailUrl = $directory.$thumbnailName;
    $article->thumbnail = $thumbnailUrl;
}
```