

Implementing the erag/laravel-pwa package

Implementing the

`erag/laravel-pwa` package involves a few quick steps using Composer and Blade directives.

Prerequisites

- A Laravel application (version 8.x to 12.x supported).
- Composer for package installation.
- An **HTTPS connection** in production for PWA features to work correctly.

Step-by-Step Implementation

1. Install the Package via Composer

Open your terminal and run the following command in your Laravel project's root directory:

```
bash
```

```
2. composer require erag/laravel-pwa
```

3. Run the Installation Command

Publish the necessary configuration files and assets using the provided Artisan command:

```
bash
```

```
4. php artisan erag:install-pwa
```

This command creates a `config/pwa.php` file and the required manifest/service worker files.

5. Configure Your PWA

Customize the `config/pwa.php` file to match your application's branding and requirements. Key elements you can modify include the app name, short name, colors, and icon paths.

Example configuration snippet:

```
php
```

```
6. // config/pwa.php
return [
    'manifest' => [
```

```
'name' => 'Your App Name',
'short_name' => 'App',
'background_color' => '#FFFFFF',
'theme_color' => '#000000',
// ... other options
],
// ...
];
```

Make sure the `public/` folder is writable.

7. Update Your Layout Files

To integrate the PWA functionality, add the provided Blade directives to your main layout file (e.g., `resources/views/layouts/app.blade.php`).

- Place `@PwaHead` inside the `<head>` tag to include meta tags and manifest links:
html

```
8. <head>
    @PwaHead
    <title>Your App Title</title>
    <!-- Other head elements -->
</head>
```

9. Place `@RegisterServiceWorkerScript` just before the closing `</body>` tag to register the service worker:

html

```
10. • <body>
    <!-- Your application content -->
    @RegisterServiceWorkerScript
</body>
```

11. Update the Manifest

After making configuration changes, especially the icons or name, run this command to update the manifest file in your public directory:

bash

```
php artisan erag:pwa-update-manifest
```

Testing Your PWA

- **Deployment:** For local testing, use a development environment that supports HTTPS, such as Laravel Valet or Laravel Homestead.
- **Audit:** Use the Lighthouse tool in Chrome DevTools to audit your PWA and ensure all requirements are met.

Once these steps are completed, your Laravel application is PWA-enabled, and users will be prompted to "Add to Home Screen" on compatible devices

Revision #1

Created 13 December 2025 18:52:54 by AI Channel

Updated 13 December 2025 18:55:10 by AI Channel