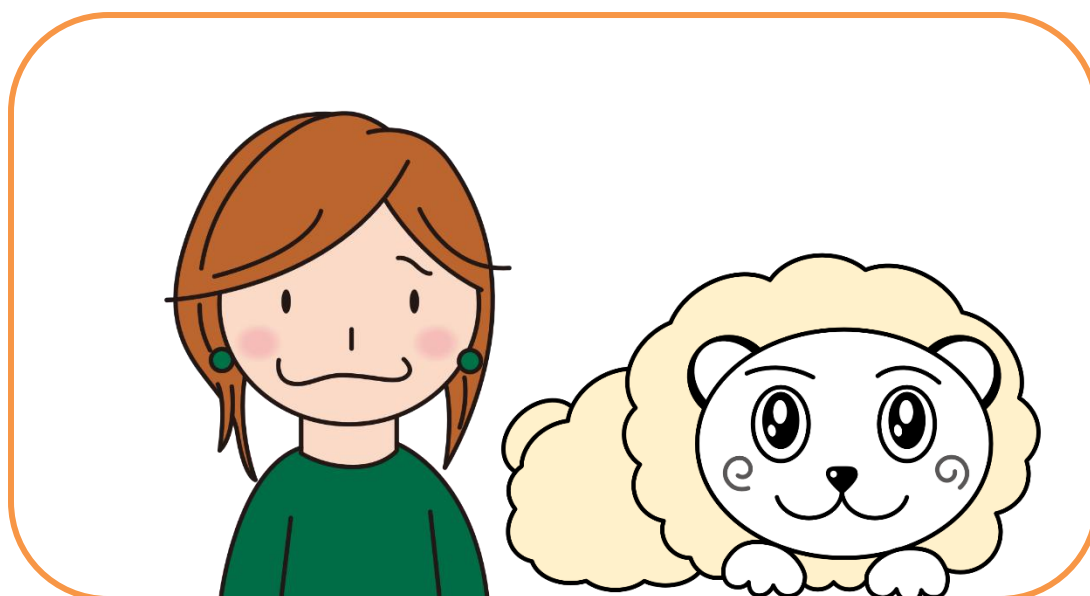


書籍【Laravel の教科書バージョン 12】

Laravel 13 で使用する場合のサポートガイド



Version 2.0

作成日 : 2026 年 4 月 3 日

内容

はじめに	4
CHAPTER 2 に関する変更点	5
PHP バージョンの要件	5
プロジェクト作成時の質問変更	6
CHAPTER 3 に関する変更点	8
ディレクトリ構造の変更 (p.92-93)	8
CHAPTER 4 に関する変更点	9
Livewire の構造 (p.108-111)	9
ルート設定の変更	9
Livewire のコードを書いてみる (p.115-118)	9
レイアウトファイルの変更	11
Volt について (p.120-122)	12
CHAPTER 5 に関する変更点	13
レイアウトの呼び出し方法の変更	13
Components について (p.142-148)	14
ログイン後ページの変更方法 (p.129)	14
CHAPTER 7~11 に関する変更点	15
ビューファイルのレイアウトタグ	15
属性構文の書き方 (ご対応不要)	15

CHAPTER 12 に関する変更点	16
make:livewire コマンドの変更 (p.324, p.334)	16
ルート設定の書き方が変わった (p.335)	16
Laravel 13 の新機能	17

はじめに

本サポートガイドは、書籍「Laravel の教科書バージョン 12」（以下【本書】とします）を、Laravel 13 環境でご利用いただく際の変更点・読み替え箇所をまとめたものです。

本書出版後に行われた主に以下の変更が含まれています。

1. **スターターキットに Fortify が組み込まれた（認証処理の変更）**
2. **Livewire 4 になった（レイアウト記法・コマンド・ファイル配置の変更）**

本書の基本的な内容（ルート設定、コントローラ、モデル、ビュー、CRUD 処理など）はそのままご利用いただけます。

本サポートガイドでは、本書の章の順番に沿って変更点をご説明していきます。

～★～☆彡～★～☆彡～★～☆彡～★～☆彡～★～☆彡～★～☆彡～★～☆彡～



「Laravel はどんどん進化していくよね。
ついていくのが大変だなあ。」



「たしかに。それだけ今の時代に合わせて進化しているってことなんだ。Laravel13
では生成 AI 関連の機能も充実してきているよ。本書の最後に少し紹介するね。」

CHAPTER 2 に関する変更点

PHP バージョンの要件

Laravel 13 では、PHP 8.3 以上が必要となります。 下記の Laravel13 インストールマニュアルに記載されている PowerShell コマンドを実行すると、PHP 8.3 以上がインストールされます。

【Laravel13 インストールマニュアル】

<https://laravel.com/docs/13.x/installation>

■ Windows の場合（PowerShell を管理者として実行）：

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object  
System.Net.WebClient).DownloadString('https://php.new/install/windows/8.4'))
```

■ Mac の場合（ターミナルで実行）：

```
/bin/bash -c "$(curl -fsSL https://php.new/install/mac/8.4)"
```

インストール完了後、以下のコマンドで PHP のバージョンを確認します。

```
php -v
```

PHP 8.3 以上が表示されれば問題ありません。もし古いバージョンが表示される場合は、過去に別の方法（XAMPP や Homebrew 等）でインストールした PHP が優先されている可能性があります。（環境変数（PATH）の設定見直し等が必要になります）。

プロジェクト作成時の質問変更

プロジェクトを作成する際には何点か質問がされます。変更点が2点あります。

```
PS C:\Users\jun\Documents\doclaravel> laravel new test

L A R A V E L

Which starter kit would you like to install? [None]:
[none ] None
[react ] React
[svelte ] Svelte
[vue ] Vue
[livewire] Livewire
> livewire

Which authentication provider do you prefer? [Laravel's built-in authentication]:
[laravel] Laravel's built-in authentication
[workos ] WorkOS (Requires WorkOS account)
[none ] No authentication scaffolding
> laravel

Would you like to use single-file Livewire components? (yes/no) [yes]:
> no

Which testing framework do you prefer? [Pest]:
[0] Pest
[1] PHPUnit
> 0

Do you want to install Laravel Boost to improve AI assisted coding? (yes/no) [yes]:
> yes
```

■ 本書 p.45 質問③ "Would you like to use Laravel Volt?" が "Would you like to use single-file Livewire components?" になりました。 **No** を選んでください。

■ 最後に "Do you want to install Laravel Boost to improve AI assisted coding?" という質問が追加されました。 **お好みですが、Laravel Boost は便利なので、もし生成 AI を使って開発をする場合には、インストールをおすすめします。**

【Laravel Boost について(筆者ブログ記事)】

<https://biz.addisteria.com/laravel-boost/>



「Laravel Boost をいれると、何がいいの？」



「生成 AI ツールがログを読み取ってくれたり、データベーステーブルを操作できるようになるよ。」



「へえ、便利そう！」



「ただ、最初は Laravel Boost なしで、全部自分で操作してみてもいいかもね。お好みでいれてね。」

CHAPTER 3 に関する変更点

ディレクトリ構造の変更 (p.92-93)

本書で説明しているディレクトリ構造から、以下の点が変わっています。

■ 認証関連ファイルの場所

スターターキットに Fortify が組み込まれたことにより、認証処理のファイルが `app/Actions/Fortify` の中に移りました。たとえば新規ユーザー登録用の処理は、変更前は `app/Livewire/Auth/Register.php` に書かれていましたが、変更後は `app/Actions/Fortify/CreateNewUser.php` に書かれています。

本書の記載	Laravel 13 での読み替え
<code>app/Livewire/Auth/</code> (<code>Register.php</code> など)	<code>app/Actions/Fortify/</code> (<code>CreateNewUser.php</code> など)

■ ビューファイルのパス

Livewire 4 の変更により、レイアウトファイルと認証ビューファイルの場所が変わりました。

本書の記載	Laravel 13 での読み替え
<code>resources/views/components/layouts/</code>	<code>resources/views/layouts/</code>

本書の記載	Laravel 13 での読み替え
<code>resources/views/livewire/auth/</code>	<code>resources/views/pages/auth/</code>

CHAPTER 4 に関する変更点

Livewire の構造 (p.108-111)

本書ではユーザー登録用ファイル (app/Livewire/Auth/Register.php) を使って Livewire の構造を説明しています。Laravel 13 では、このファイルはなくなりました。ユーザー登録の処理は Fortify が担っています。

ユーザー登録用のビューファイルの場所も変わっています。

本書の記載	Laravel 13 での読み替え
resources/views/livewire/auth/register.blade.php	resources/views/pages/auth/register.blade.php

また、ビューファイル内の wire コードがなくなり、通常の Laravel の書き方に沿って、form タグ内の action 属性にフォーム送信先のルートが指定されています。本書 p.108-111 の構造の説明は、上記を踏まえてお読みください。

ルート設定の変更

これまでは、routes/auth.php にユーザー認証用のルート設定がはいつていました。今回の変更により、この routes/auth ファイルがなくなりました。ユーザー認証用のルート設定は、vendor ディレクトリ以下 (vendor/laravel/fortify/routes/routes.php) に入っています。**このファイルを編集する必要はありません。**

Livewire のコードを書いてみる (p.115-118)

本書ではユーザー登録用ファイルを使って Livewire のコードを試していますが、Fortify 導入により、ユーザー登録部分の処理に Livewire が使用されなくなりました。そのため、p.115-118 の Livewire 処理については、ユーザー登録用ファイルではなく、**プロフィールページを使用してお試しください。**プロフィールページは、resources/views/pages/settings/profile.blade.php となります。

【115-116 ページコード・変更後（赤色部分を追加）】 profile.blade.php

(省略)

```
<form wire:submit="updateProfileInformation" class="my-6 w-full space-y-6">
  <flux:input wire:model.live="name"
    :label="__('Name')"
    type="text"
    required autofocus autocomplete="name"
  />
  入力中 : {{ $name }}
```

【116-117 ページコード・変更後（赤色部分を追加）】 profile.blade.php

(省略)

```
<form wire:submit="updateProfileInformation" class="my-6 w-full space-y-6">
  <flux:input wire:model.live.debounce.1000ms="name"
    :label="__('Name')"
    type="text"
    required autofocus autocomplete="name"
  />
  入力中 : {{ $name }}
```

次に、p.117 からご紹介している[4-3-2 リアルタイムバリデーション]については、プロフィールページを下記のように変更します。

【117 ページコード・変更後（赤色部分を追加）】 profile.blade.php

(省略)

```
<form wire:submit="updateProfileInformation" class="my-6 w-full space-y-6">
  <flux:input wire:model.blur="name"
    :label="__('Name')"
    type="text"
    required autofocus autocomplete="name"
  />
```

処理部分については app/Livewire/Settings/Profile.php の中の updatedName メソッドを下記のように追加します。ユーザー名が 2 文字以上の場合はエラーとなります。

【117-118 ページコード・変更後（赤色部分を追加）】 Profile.php

（省略）

```
public function updatedName(): void
{
    $this->validateOnly('name', [
        'name' => ['max:2'],
    ]);
}
```

また、ビュー側の `wire:model.blur` を `wire:model.blur.live` に変更してください。

レイアウトファイルの変更

変更前は、ビューファイルのレイアウトの指定は、`app/Livewire/Auth/Register.php` に次のように記述されていました。

```
#[Layout('components.layouts.auth')]
```

前述のとおり、変更後は、`app/Livewire/Auth/Register.php` がなくなりました。レイアウトの指定は、`resources/views/pages/auth/register.blade.php` ファイルの最初と最後で、タグを使って次のように行われています。

```
<x-layouts::auth>
（省略）
</x-layouts::auth>
```

レイアウトとして使用されるファイルは、`resources/views/layouts/auth.blade.php` です。

Volt について (p.120-122)

本書では Volt という名称でご紹介していますが、Livewire 4 では Volt は廃止され、「**シングルファイルコンポーネント**」に名前が変わりました。Livewire の公式マニュアルには次のように書かれています。

“Livewire v4 now supports single-file components, which use the same syntax as Volt class-based components.”

「Livewire v4 では、Volt のクラスベースのコンポーネントと同じ構文を使用するシングルファイルコンポーネントがサポートされるようになりました。」

つまり Volt の名称はなくなりましたが、機能自体は残り、シングルファイルコンポーネントに引き継がれています。Livewire のコンポーネントとビューは一つのファイルに書くこともできますし、別々に書くことも可能です。本書では別々に書く方法をご紹介します。別々に書くためには、以下の 2 点が必要です。

1. プロジェクト作成時の質問 "Would you like to use single-file Livewire components?" に No を選択
2. make:livewire コマンドに --class オプションをつける (本ガイド CHAPTER 12 にて説明)



CHAPTER 5 に関する変更点

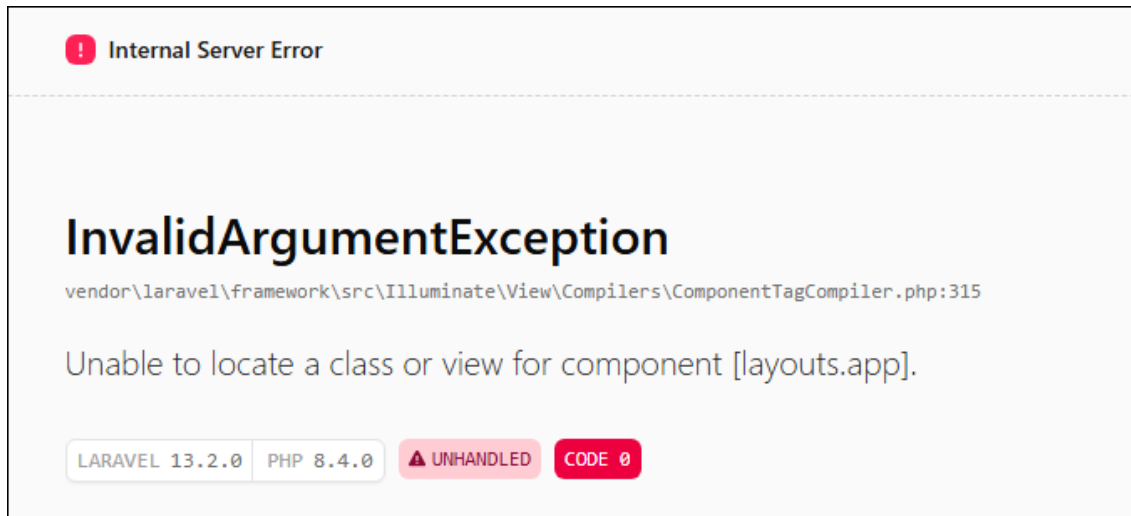
レイアウトの呼び出し方法の変更

Livewire 4 では、レイアウトの呼び出し記法が変わりました。layouts の後ろのドット (.) がダブルコロロン (::) に変わります。

本書の記載	Laravel 13 での読み替え
<code><x-layouts.app></code> <code></x-layouts.app></code>	<code><x-layouts::app></code> <code></x-layouts::app></code>

本書の記載	Laravel 13 での読み替え
<code><x-layouts.auth></code> <code></x-layouts.auth></code>	<code><x-layouts::auth></code> <code></x-layouts::auth></code>

本書の中で、`x-layouts.app`、`x-layouts.auth` とある部分については、上記のように読み替えてください。また、新たにコードを入力する際にも、上記のように `layouts` の後にはダブルコロロンを入れるようにしてください。もしダブルコロロンにしていな場合には、下記のようにエラーメッセージが表示されます。



Livewire4 については、下記公式マニュアルもご参照ください。

【Livewire4 公式アップグレードガイド】

<https://livewire.laravel.com/docs/4.x/upgrading>

Components について (p.142-148)

本書で説明しているコンポーネント関連のファイルパスを読み替えてください。

本書の記載	Laravel 13 での読み替え
resources/views/components/layouts/app.blade.php	resources/views/layouts/app.blade.php

本書の記載	Laravel 13 での読み替え
resources/views/components/layouts/auth.blade.php	resources/views/layouts/auth.blade.php

本書の記載	Laravel 13 での読み替え
resources/views/components/layouts/app/sidebar.blade.php	resources/views/layouts/app/sidebar.blade.php

レイアウトファイルの中身や役割は変わっていません。パスだけの変更です。

ログイン後ページの変更方法 (p.129)

本書では app/Livewire/Auth/Register.php 内の redirect 先を変更する方法を説明していますが、Laravel13 ではこのファイルはありません。代わりに、**config/fortify.php ファイルの中の 'home' の値を変更してください。**

```
'home' => '/dashboard',
```

この '/dashboard' の部分を、表示したいページのパスに変更します。

CHAPTER 7～11 に関する変更点

CHAPTER 7 から CHAPTER 11 の内容（フォーム作成、データ保存、一覧表示、リレーション、ミドルウェア、Gate、CRUD、テストデータ、ページネーション、コンポーネント作成）については、以下の 1 点を除き、本書の通りに進められます。

ビューファイルのレイアウトタグ

新しくビューファイルを作成する際（`create.blade.php`、`index.blade.php`、`show.blade.php`、`edit.blade.php` など）、レイアウトの指定は以下のように記述してください。

本書の記載	Laravel 13 での読み替え
<code><x-layouts.app></code>	<code><x-layouts::app></code>

属性構文の書き方（ご対応不要）

Laravel 13 では、PHP の属性構文（Attributes #[]）を使った記述が増えました。これは PHP 8.0 で導入された書き方で、クラスやプロパティ等に設定情報を付与するものです。

■これまでの書き方例

```
protected $fillable = ['name', 'email', 'password'];
```

■Laravel 13 からの新しい書き方例

```
#[Fillable(['name', 'email', 'password'])]
```

ただ、以前の書き方でも動きます。Laravel 13 のデフォルトのファイルでは、上記の新しい書き方になっています。

CHAPTER 12 に関する変更点

make:livewire コマンドの変更 (p.324, p.334)

Livewire 4 では make:livewire コマンドのデフォルト動作が変わりました。何もオプションをつけ
ないと、コンポーネントとビューファイルがひとつになり resources/views/components/ の中に
ファイルが作られます。

本書のようにコンポーネントクラスとビューを分けてファイルを作成する場合には、**--class オプションをつけてください。本書と同じ場所にファイルが作られます。**

■ 324 ページ

```
php artisan make:livewire PostFilter --class
```

■ 334 ページ

```
php artisan make:livewire UserList --class
```

ルート設定の書き方が変わった (p.335)

Livewire 4 では、Livewire コンポーネントのルートは Route::livewire() で書くのが推奨です。

【書籍】

```
Route::get('/users', UserList::class)->name('users.list');
```

【Laravel 13】

```
Route::livewire('/users', UserList::class)->name('users.list');
```

ただ、書籍の書き方 (Route::get) でも動作します。

Laravel 13 の新機能

Laravel 13 では、生成 AI を便利に使える新機能「Laravel AI SDK」も追加されました。こちらは別途インストールが必要となります。

機能としては、「Web アプリに AI を組み込むための公式の工具箱」といったもの。ChatGPT など、生成 AI を Web アプリに組み込む手間が大幅に減らせます。

Laravel13 の新機能・変更点にご興味あれば、筆者のブログ・記事もご参照ください。

【Laravel13 新機能・変更点 (筆者ブログ記事)】

<https://biz.addisteria.com/laravel-13/>

【Youtube 動画版】

https://youtu.be/a9jjLmUegIw?si=O2_yloRsrPAICI6I



「生成 AI を使って Web アプリをつくるって楽しそう！」



「できることがまたグッと広がるよ。まずは Laravel の教科書で基本をおさえて、その後色々な Web アプリ作成にチャレンジしてね。」