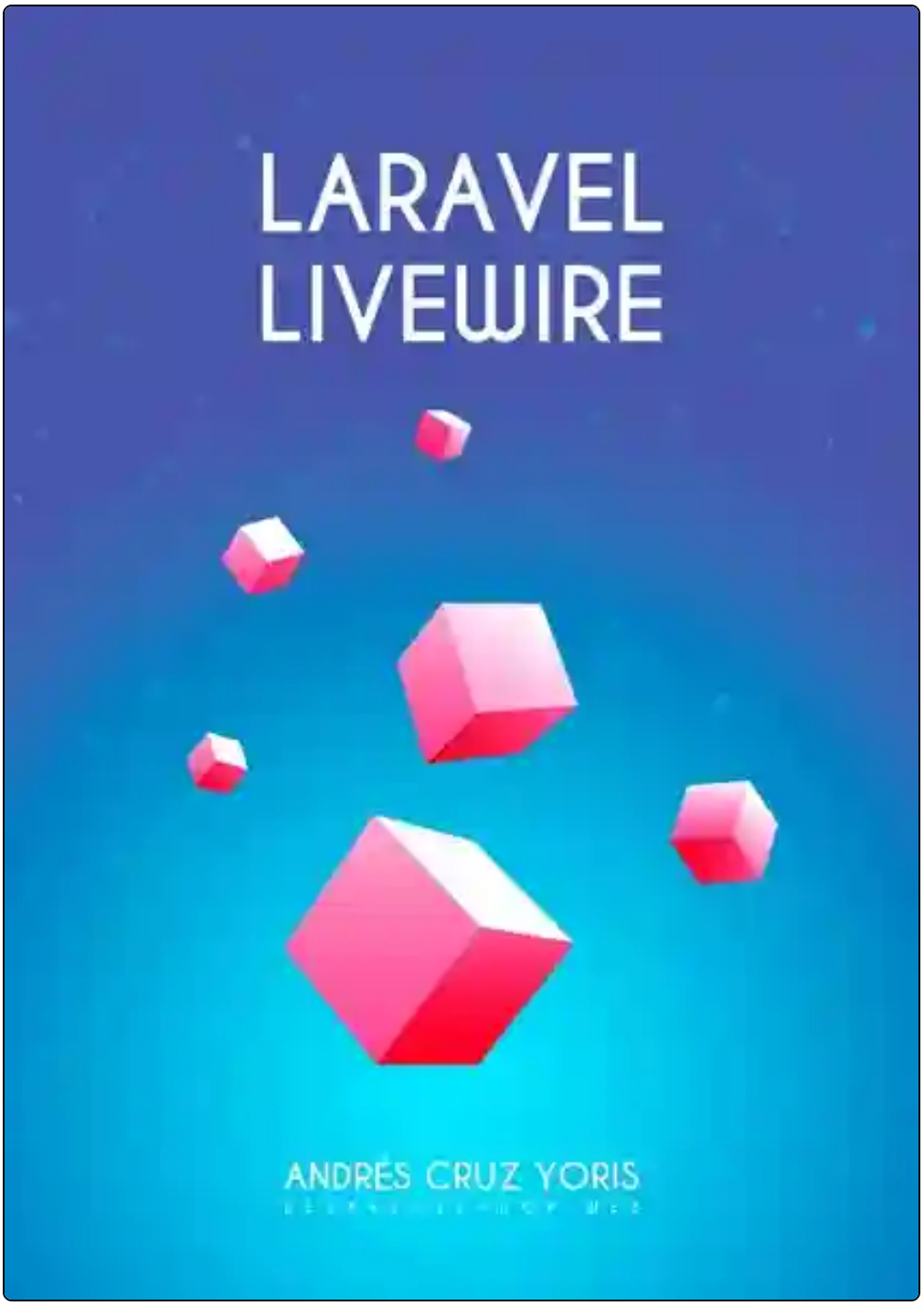

LARAVEL
LIVEWIRE
ANDRÉS CRUZ YORIS

Getting started with Laravel 12 Livewire 3

Here continue your roadmap in the development of web applications in Laravel with Livewire, volt and flux

Andres Cruz Yoris

This version was published: 2025-05-02

Posted about the book!

Please help promote this book.
The suggested post for this book is:
I just bought the book "Getting started with Laravel Livewire" from @LibreDesarrollo!

Get your copy at:

https://www.desarrollolibre.net/libros/primeros-pasos-laravel-livewire

About the Author

This book was prepared by Andres Cruz Yoris, Bachelor of Computer Science, with more than 10 years of experience in the development of web applications in general; I work with PHP, Python and client-side technologies such as HTML, JavaScript, CSS, Vue among others; and server-side like Laravel, Flask, Django, and CodeIgniter. I am also a developer in Android Studio, xCode and Flutter for the creation of native applications for Android and IOS.

I put at your disposal part of my learning, reflected in each of the words that make up this book, my second book on Laravel; in this case more advanced and has been the development with Laravel Livewire.

Copyright

No part of this book may be reproduced or transmitted in any way; that is, electronically or by photocopies without the permission of the author.

Foreword

Laravel is a fascinating framework, huge and with a somewhat steep learning curve and with multiple options; this book is not for beginners and assumes that you already know how to program in Laravel.

Laravel Livewire takes Laravel development a step further; automating routine processes in very flexible schemes based on components, Livewire components that allow us to communicate client and server in a very simple, efficient and easy way.

Laravel Livewire is not a framework, it is just a layer or scaffolding that adds certain extra features to the framework that we can use to create great applications, with less effort and development time.

Who is this book for

This book is aimed at anyone who wants to start developing with Laravel Livewire in developing applications based on Laravel components with some additions that allow direct communication between server and client.

This book is not recommended for those people who have not worked with Laravel, if this is your case, I advise you to first know and practice with Laravel before starting; on my website you will find courses, books, publications, videos and in general more information about Laravel.

For people who want to improve a skill in web development, who want to grow as a developer.

As long as you identify with at least one of the points mentioned above, this book is for you.

Considerations

Remember that when you see the $ symbol it is to indicate commands in the terminal; you do not have to type this symbol in your terminal, it is a convention that helps you know that you are executing a command.

At the end of each chapter, you have the link to the source code so you can compare it with your code.

The use of bold in paragraphs has two functions:
If we place a card in bold it is to highlight some code such as the name of variables, tables or the like.
Highlight important parts or ideas.

To display tips we use the following layout:

important tips

For the code snippets:

```
$ npm run watch
```

When using the *** or //***

It means that we are indicating that in the code there are fragments that we presented earlier.

The official page of the framework has been:

https://laravel.com/

From the project structure:

https://jetstream.laravel.com/2.x/stacks/livewire.html

From what Laravel Livewire offers us when developing:

https://laravel-livewire.com/

Which is essential to follow this book and with this learn more aspects of the framework.

Errata and comments

If you have any questions, recommendations or found any errors, you can let them know at the following link:

https://www.desarrollolibre.net/contacto/create

By my email:

desarrollolibre.net@gmail.com

Or by Discord on the Laravel channel:

https://discord.gg/sg85Zgwz96

As a recommendation, before reporting a possible problem, check the version to which you refer and add it in your comment; it is found on the second page of the book.

The book was originally written in spanish by the author; any problem with the translation, you can please communicate it through the channels presented.

Introduction

This guide is intended to get you started with Laravel Livewire; with this, we are going to pose two things:
It is not a book that aims to know 100% Laravel Livewire, the main objective is to know its main components based on examples.
It is assumed that the reader has at least basic knowledge of Laravel.

Laravel, being a more complete framework that you want to deal with, in practice, has many more components to work with, it is assumed that the reader has some basic knowledge about how the framework works, such as the use or theory of what migrations, MVC, routes, among others, work for; you don't need to know how to handle them, but you do need to understand the logic behind it all; if you don't have them, I recommend that you see my first web programming book in which we take the first steps with CodeIgniter, which is a great framework with many similarities with Laravel, and being a smaller framework and easier to handle is more easy to start your learning.

Finally; compared to other books, the approach will be a little more accelerated or general when it comes to the explanations of the elements that make up the framework; and this is so for two main aspects:
I want to cover as many of the features of Laravel Livewire without making the book too long.
This is not a recommended book if this is the first PHP framework of this type that you are facing, therefore, being so, you should already know these aspects of the structure of the framework.

To follow this book you need to have a computer running Windows, Linux, or MacOS.

Remember that the book is currently in development.

Map

Chapter 1: In this chapter we are going to give a brief introduction about Laravel Livewire.

Chapter 2: In this chapter we are going to create our project in Laravel Livewire.

Chapter 3: In this chapter, we present the characteristics that a project created in Laravel Livewire has; register, and authentication, are some of them.

Chapter 4: In this chapter, we are going to create a simple CRUD type application and get to know all the elements of Livewire such as handling forms, views and directives through the Livewire components.

Chapter 5: In this chapter, we are going to work on the style, improving the created application so that it looks like another Laravel Livewire module.

Chapter 6: In this chapter, we are going to learn about the communication between Livewire components based on events; detect changes in properties of the component class from the view and know the Laravel Livewire JavaScript

Chapter 7: In this chapter, we are going to create the CRUD for the posts.

Chapter 8: In this chapter, we are going to create filters and search fields for a listing taking advantage of the benefits of Livewire.

Chapter 9: In this chapter, we are going to create sort fields for the column listing.

Chapter 10: In this chapter, we will get to know everything that Livewire offers us in JavaScript and combine it with other plugins, we will know the hooks in JavaScript, consume properties, functions and use of events.

Chapter 11: In this chapter, we are going to learn about some extra features of Laravel Livewire properties, such as customizing the update phase of wire:models, such as the use of computed properties.

Chapter 12: In this chapter, we are going to create a simple blog application with a list with filters and search and detail fields.

Chapter 13: In this chapter, we are going to create a shopping cart in the session and database, for which, we will see communication between components, sending events, toast type plugin to show messages confirming the operation carried out, with listeners of the events from JavaScript.

Chapter 14: In this chapter, we are going to show the code of an application created with Alpine.js to create a to do list component.

Chapter 15: In this chapter, we are going to adapt the Alpine.js application presented in the previous chapter, to work in conjunction with Livewire.

Chapter 16: We will learn how to create unit and integration tests in the application we created previously using PHPUnit.

Chapter 17: In this chapter, we present the characteristics that a project created in Laravel Livewire has; team management, API Tokens, and authentication, are some of them.

Index

- **Chapter 1: About Laravel Livewire**
  - What does Livewire offer us?
    - Livewire Volt and Flux

- **Chapter 2: Create a project**
  - Create the project in Laravel Livewire on MacOS or Linux
  - Create the project in Laravel Livewire on Windows
  - Second way to install Laravel Livewire

- **Chapter 3: Characteristics of a Livewire project**

- **Chapter 4: Introduction to Livewire Components**
  - Migrations
  - Models
  - Components
  - Define routes
  - Extra: Error not found: [components.layouts.app]
    - Publish configuration file
    - Generate a layout
  - Testing with a Component
  - Component to create a category
  - Wire:x directive references
  - Requests to the server from Livewire
  - wire:blur
  - Validate forms
    - Case study
  - Reuse Livewire Components
  - Life cycle of a component
  - Update a record
  - List
    - Paginated list
  - Delete

        - Delete with confirmation dialog
        - Warning message
        - Upload files (upload the image for the category)

- **Chapter 5: Components and Styles**
    - Action taken message
    - Menu
    - Button to create
    - Layout header
    - Alert Dialog to Delete: Confirmation Component
    - Extra: Translate to volt components
    - Extra: Wire:confirm

- **Chapter 6: Communication between components**
    - Create migrations
    - Livewire Components
    - Models
    - Migrations
    - Form to create a general contact
        - Validations using attributes
    - Form to create a contact per person
    - Form to create a contact by company
    - Form to create a contact by detail
    - Initial form logic step by step
    - @livewire directive to render components
    - Events: Communication between components
        - Case study
    - Define layout
    - Step-by-step design
    - $wire object
        - Update the step by step with the $step property
    - Step 3:
        - Last step
    - Parent to child communication: Send the general contact id to child components
    - Enable saving on child components
    - Load registered contact data
    - Passing parameters from parent to children
    - Update existing general contact
    - Final styling details

- Extra: Translate to volt components

- **Chapter 7: CRUD for posts**
  - General changes
  - Save and edit
  - List
  - Extra: Translate to volt components
  - Form object

- **Chapter 8: Filters and search fields**
  - Table changes
  - Seeders for posts
  - Filters for categories, posts and types
  - Search field for id, title and description
  - Filter by date range
  - Query String: Parameters in the URL
  - Filters with conditional clauses, when
  - Form object
  - Clean filter

- **Chapter 9: Sort fields**
  - Traits and Livewire components for code reuse
  - Clear filter
  - Extra: Translate to volt components

- **Chapter 10: Livewire JavaScript**
  - Introduction to Hooks in JavaScript
    - morph.removed(ing) hook
    - morph.updated(ing) hook
    - Listen to events through messages
  - Accessing properties from JavaScript and CKEditor
    - CKeditor
  - Resume
  - Extra: Translate to volt components

- **Chapter 11: Livewire Properties**
  - Defer update for wire:model
    - .debounce: Modify the update time
    - .lazy: Send updates when losing focus

        - .defer: Disable updates
    - Computed properties
    - Blocked Properties
    - Resetting properties
    - Supported property types

- **Chapter 12: Simple Blog Web**
    - Routes
    - Model
    - Layout for the web
    - Card component
    - Post listing
        - Replace link with Flux button
    - Post detail
        - Add step by step to detail view
    - Extra: Detailed links to the listing by type and category
    - Extra: Translate to volt components

- **Chapter 13: Shopping cart**
    - General scheme
    - Item component: Managing an item in the cart
        - Delete cart session
    - Changes in the Cart.php component
        - View to add a post
    - Replicate the shopping cart in the database
        - Create migration and table
        - Model
        - Record changes to the database
        - Replicate from database to session on login
    - Show a message when performing actions on the cart
        - Broadcast events
        - Install and configure toast and listen for events
    - Button to show the total number of items
        - Show total and link to list of cart items
    - Improve design
        - Remove option to add the product in detail if it is already in the cart
    - Reload list
        - Review of the events
        - Key Component: Force component update

- Animated transition
- Extra: Translate to volt components

- **Chapter 14: Create an Alpine.js to do list application**

- **Chapter 15: Adapting the Alpine.js To do list Application with Livewire**
  - Create component in Livewire
    - Install and configure dependencies
  - Create migration and model for Todos
  - Create to do from Livewire
  - To Do List
  - Replicate sort to do property
  - Replicate sort in the database
  - Ordination, second scheme
    - Problem synchronizing added tasks
  - Ordination, third scheme
  - Delete a to do
  - Delete all to do
  - Edit a to do
  - Mark completed a to do
  - Extra: Reordering of to do by axios
  - Migrate from Bootstrap to Tailwind
  - Validations

- **Chapter 16: Testing**
  - First steps
    - From Laravel to Livewire
  - Blog
    - Test for the List
    - Test for detail
    - Test for the filter
  - Configure database for testing
  - Authentication
  - Dashboard
    - Category
    - Post
  - Step by step form
    - First step: General
    - Second step: Company

        - Second step: Person
        - Third step: Detail
        - assertDispatched
        - Edit
        - Validation errors
    - Cart
    - To Do
    - Upload
    - Where to go from here:
    - Conclusions

- **Chapter 17: Legacy: Characteristics of a Livewire project with Jetstream**
    - Laravel Fortify
    - Options
        - Options in profile
        - Options in API Tokens
        - Teams options
    - Project structure
    - Livewire Components

Preview

This is a Preview of the book content and it is completely FREE, you can get the full book from: **desarrollolibre.net/libros**.

Chapter 1: About Laravel Livewire

Livewire is a scaffolding for Laravel, which is nothing more than a skeleton or layer that adds certain extra features in Laravel through the use of Livewire components; the same basic Laravel components, but with very interesting additions that are what make up Livewire.

In addition to this, a project in Livewire already brings us Tailwind.css and Alpine.js configured.

In this book we will use the term "Basic Laravel" to mean the Laravel framework without any add-ons or scaffolding as is the case with Laravel Livewire.

What does Livewire offer us?

Building modern web apps is hard, tools like Vue and React are extremely powerful, but the complexity they add to a full-stack developer's workflow is insane.

With the above in mind, Laravel Livewire is a complete framework for Laravel that makes it easy to create dynamic interfaces, without leaving the convenience of Laravel; in short, it allows us to use schemes similar to those of Vue and React directly in Laravel.

We can work with schemes similar to those of Laravel and Vue, but in a more direct and simple way; Laravel development is heavily tied to the use of components; the same Laravel components but with important additions from Laravel Livewire which allows you to greatly scale the use of components:

1. Link using the routes file.
2. Simple communication based on actions, events and functions between the client and the server.
3. Add features such as pagination, file upload, query string among others.

In short, with Livewire, we can do the same thing that we do with Laravel and its controllers, but in a simpler way in which we can use the components and reuse pieces of code that Livewire offers us to make a very simple communication between the server with the client.

Apart from that, depending on how you decide to install Laravel Livewire, you can enable options that come for free like:

1. Authentication system with registration, credential recovery.
2. Profile view with user upload.

3. Management of roles through teams.
4. Handling the Tokens API using Laravel Breeze with a manageable interface.

In short, Livewire is not a framework, you can see it as another package that adds extra features to some Laravel elements that ultimately make it a scaffolding or skeleton for our applications.

For this technology, we have to use one separate documentation (not counting Laravel).

The one that Laravel Livewire offers us when developing:

https://laravel-livewire.com/

Livewire Volt and Flux

In recent Laravel Livewire projects, starting with Laravel version 12, when creating a project in Laravel Livewire it includes other technologies such as:

Volt:

https://livewire.laravel.com/docs/volt

With which, we can create both the logic and the view (blade) in a single file:

```php
<?php

use function Livewire\Volt\{state};

state(['count' => 0]);

$increment = fn () => $this->count++;

?>

<div>
    <h1>{{ $count }}</h1>
    <button wire:click="increment">+</button>
</div>
```

Flux, which is nothing more than a library of ready-to-use components for Laravel Livewire:

https://github.com/livewire/flux

```
<flux:input
    wire:model="email"
    :label="__('Email address')"
    type="email"
    required
    autocomplete="email"
    placeholder="email@example.com"
/>
```

What you can publish to personalize:

```
$ php artisan flux:publish
```

Chapter 2: Create a project

It is assumed that the reader has the necessary knowledge to develop in basic Laravel; being so, it will be done for a mere mention of the environment that will be used to develop in Laravel, which will be:

- Laravel Sail with Docker for Linux and MacOS.
- Laragon for Windows.
- Laravel herd for Windows and MacOS.

Although you are free to use another of the many available for Laravel.

Create the project in Laravel Livewire on MacOS or Linux

In this section, we are going to create the project with which we are going to work throughout the book for MacOS or Linux.

In recent versions, the installer asks you first if you want to create a project with Livewire:

```
$ laravel new livewirestore
```

A screen like the following will appear:

```
 _                              _
| |                            | |
| |    __ _ _ __ __ ___   _____| |
| |   / _` | '__/ _` \ \ / / _ \ |
| |___| (_| | | | (_| |\ V /  __/ |
|______\__,_|_|  \__,_| \_/ \___|_|

Which starter kit would you like to install?

? None
? React
? Vue
? Livewire
```

Select:

```
Livewire
```

Which Jetstream stack do you prefer?

```
[0] livewire
[1] inertia
```

Next, it will ask you which authentication system you want to use:

```
Which authentication provider do you prefer? [Laravel's built-in authentication]:
 [laravel] Laravel's built-in authentication
 [workos ] WorkOS (Requires WorkOS account)
```

You can select any of them, in the book we selected option "1".

Once the process is finished, so that it can work with Laravel Sail (if you are going to use Sails and not Laragon/Herd); whose package is installed by default:

```
$ php artisan sail:install
```

We get up the Sail process; so that it generates the image in Docker, configure our project, and in a few words, we have everything ready:

```
$ ./vendor/bin/sail up
```

With this, we will see in Docker:

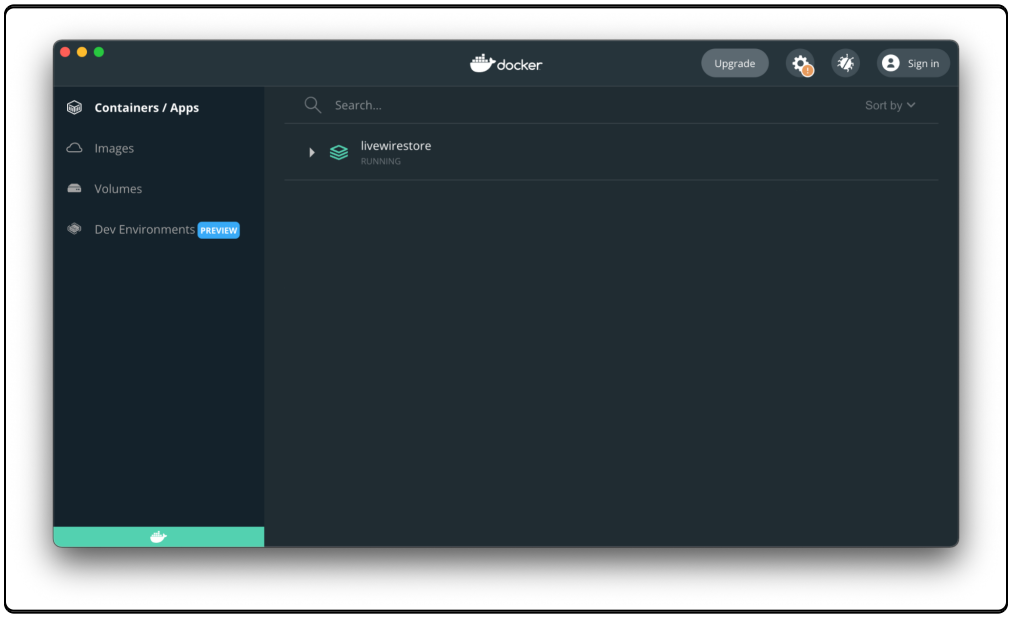


Figure 2-1: Laravel Livewire in Docker

Remember that you should not have any other process started on port 80; eg Apache; would give an error like the following:

Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:80 -> 0.0.0.0:0: listen tcp 0.0.0.0:80: bind: address already in use



Figure 2-2: Localhost

To solve it, you must stop the process that has that port occupied; in the case of Apache, it would be something like:

...

Chapter 3: Characteristics of a Livewire project

In this chapter, we'll discuss the most important aspects of creating a project in Laravel Livewire starting with Laravel version 12. As we mentioned before, unlike previous versions, Laravel no longer includes the features we had in previous versions with Jetstream.

The first thing we see is a pair of links to register and log in. At this point, if you haven't already done so, create a user. Once registered or logged in, you'll have access to:

http://livewirestore.test/dashboard

...