

Contribution Guide

- [Bug Reports](#)
- [Support Questions](#)
- [Core Development Discussion](#)
- [Which Branch?](#)
- [Compiled Assets](#)
- [Security Vulnerabilities](#)
- [Coding Style](#)
 - [PHPDoc](#)
 - [StyleCI](#)
- [Code of Conduct](#)

Bug Reports

To encourage active collaboration, Laravel strongly encourages pull requests, not just bug reports. Pull requests will only be reviewed when marked as "ready for review" (not in the "draft" state) and all tests for new features are passing. Lingerin, non-active pull requests left in the "draft" state will be closed after a few days.

However, if you file a bug report, your issue should contain a title and a clear description of the issue. You should also include as much relevant information as possible and a code sample that demonstrates the issue. The goal of a bug report is to make it easy for yourself - and others - to replicate the bug and develop a fix.

Remember, bug reports are created in the hope that others with the same problem will be able to collaborate with you on solving it. Do not expect that the bug report will automatically see any activity or that others will jump to fix it. Creating a bug report serves to help yourself and others start on the path of fixing the problem. If you want to chip in, you can help out by fixing [any bugs listed in our issue trackers](#). You must be authenticated with GitHub to view all of Laravel's issues.

If you notice improper DocBlock, PHPStan, or IDE warnings while using Laravel, do not create a GitHub issue. Instead, please submit a pull request to fix the problem.

The Laravel source code is managed on GitHub, and there are repositories for each of the Laravel projects:

- [Laravel Application](#)
- [Laravel Art](#)
- [Laravel Documentation](#)
- [Laravel Dusk](#)
- [Laravel Cashier Stripe](#)
- [Laravel Cashier Paddle](#)
- [Laravel Echo](#)
- [Laravel Envoy](#)
- [Laravel Folio](#)
- [Laravel Framework](#)
- [Laravel Homestead \(Build Scripts\)](#)
- [Laravel Horizon](#)
- [Laravel Livewire Starter Kit](#)
- [Laravel Passport](#)
- [Laravel Pennant](#)
- [Laravel Pint](#)
- [Laravel Prompts](#)
- [Laravel React Starter Kit](#)
- [Laravel Reverb](#)
- [Laravel Sail](#)
- [Laravel Sanctum](#)
- [Laravel Scout](#)
- [Laravel Socialite](#)
- [Laravel Telescope](#)
- [Laravel Vue Starter Kit](#)
- [Laravel Website](#)

Support Questions

Laravel's GitHub issue trackers are not intended to provide Laravel help or support. Instead, use one of the following channels:

- [GitHub Discussions](#)
- [Laracasts Forums](#)
- [Laravel.io Forums](#)
- [StackOverflow](#)
- [Discord](#)
- [Larachat](#)
- [IRC](#)

Core Development Discussion

You may propose new features or improvements of existing Laravel behavior in the Laravel framework repository's [GitHub discussion board](#). If you propose a new feature, please be willing to implement at least some of the code that would be needed to complete the feature.

Informal discussion regarding bugs, new features, and implementation of existing features takes place in the [#internals](#) channel of the [Laravel Discord server](#). Taylor Otwell, the maintainer of Laravel, is typically present in the channel on weekdays from 8am-5pm (UTC-06:00 or America/Chicago), and sporadically present in the channel at other times.

Which Branch?

All bug fixes should be sent to the latest version that supports bug fixes (currently [12.x](#)). Bug fixes should **never** be sent to the [master](#) branch unless they fix features that exist only in the upcoming release.

Minor features that are **fully backward compatible** with the current release may be sent to the latest stable branch (currently [12.x](#)).

Major new features or features with breaking changes should always be sent to the [master](#) branch, which contains the upcoming release.

Compiled Assets

If you are submitting a change that will affect a compiled file, such as most of the files in [resources/css](#) or [resources/js](#) of the [laravel/laravel](#) repository, do not commit the compiled files. Due to their large size, they cannot realistically be reviewed by a maintainer. This could be exploited as a way to inject malicious code into Laravel. In order to defensively prevent this, all compiled files will be generated and committed by Laravel maintainers.

Security Vulnerabilities

If you discover a security vulnerability within Laravel, please send an email to Taylor Otwell at taylor@laravel.com. All security vulnerabilities will be promptly addressed.

Coding Style

Laravel follows the [PSR-2](#) coding standard and the [PSR-4](#) autoloading standard.

PHPDoc

Below is an example of a valid Laravel documentation block. Note that the `@param` attribute is followed by two spaces, the argument type, two more spaces, and finally the variable name:

```
/**
 * Register a binding with the container.
 *
 * @param string|array $abstract
 * @param \Closure|string|null $concrete
 * @param bool $shared
 * @return void
 *
 * @throws \Exception
 */
public function bind($abstract, $concrete = null, $shared = false)
{
    // ...
}
```

When the `@param` or `@return` attributes are redundant due to the use of native types, they can be removed:

```
/**
 * Execute the job.
 */
public function handle(AudioProcessor $processor): void
{
    //
}
```

However, when the native type is generic, please specify the generic type through the use of the `@param` or `@return` attributes:

```
/**
 * Get the attachments for the message.
 * @return array<int, \Illuminate\Mail\Mailables\Attachment>
 */
public function attachments(): array
{
    return [
        Attachment::fromStorage('/path/to/file'),
    ];
}
```

StyleCI

Don't worry if your code styling isn't perfect! [StyleCI](#) will automatically merge any style fixes into the Laravel repository after pull requests are merged. This allows us to focus on the content of the contribution and not the code style.

Code of Conduct

The Laravel code of conduct is derived from the Ruby code of conduct. Any violations of the code of conduct may be reported to Taylor Otwell (taylor@laravel.com):

- Participants will be tolerant of opposing views.
- Participants must ensure that their language and actions are free of personal attacks and disparaging personal remarks.
- When interpreting the words and actions of others, participants should always assume good intentions.
- Behavior that can be reasonably considered harassment will not be tolerated.