# Bagisto Contribution Guide

\*\*BUGS:\*\*

To encourage active collaboration, Bagisto encourages pull requests, not just bug reports. "Bug reports" may also be sent in the form of a pull request containing negative test.

However, when filing 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. Creating a bug report serves to help yourself and others start on the path of fixing the problem.

\*\*Projects that you can contribute in:\*\*

1. Bagisto

2. Bagisto docs

3. Laravel-aliexpress-dropship

4. Laravel-aliexpress-dropship-chrome-extension

5. Bagisto-custom-style-extension

\*\*Core development ideas or discussion:\*\*

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 comments of the issues filed using feature template.

\*\*Which branch you should target?\*\*

All bug fixes should be sent to the latest staging branch i.e development branch. 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 backwards compatible with the current Laravel release may be sent to the latest stable branch.

Major new features should always be sent to the master branch, which contains the upcoming Bagisto release.

\*\*Compiling assets:\*\*

If you are submitting a change that will affect a compiled file, such as most of the files in admin/resources/assets/sass or admin/resources/assets/js of the Bagisto 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 Bagisto. In order to defensively prevent this, all compiled files will be generated and committed by Bagisto maintainers.

\*\*Code style:\*\*

Bagisto follow PSR-2 for coding standard and PSR-4 as of Laravel for autoloading standards.

\*\*PHPDoc:\*\*

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

/\*\*

\* Register a service with CoreServiceProvider.

\*

\* @param string|array $loader

\* @param \Closure|string|null $concrete

\* @param bool $shared

\* @return void

\* @throws \Exception

\*/

protected function registerFacades($loader, $concrete = null, $shared = false)

{

//

}