# How to contribute

The easiest way to contribute is to open an issue and start a discussion.

Then we can decide if and how a feature or a change could be implemented and if you should submit a pull requests with code changes.

## General feedback and discussions

Please start a discussion on the [main issue tracker](https://github.com/FluentValidation/FluentValidation/issues).

## Building

Run `Build.cmd` (windows) or build.sh (Linux/mac) from the command line. This builds the project and and runs tests. Building requires the following software to be installed:

\* Windows Powershell or Powershell Core

\* .NET Core 3.1 SDK

\* .NET Core 2.2 SDK

\* .NET Framework 4.6.1+ (Windows)

## Bugs and feature requests?

Please log a new issue in the appropriate GitHub repository:

\* [FluentValidation or FluentValidation.AspNetCore](https://github.com/FluentValidation/FluentValidation)

\* [Legacy ASP.NET MVC5 / WebApi 2 extensions](https://github.com/FluentValidation/FluentValidation.LegacyWeb)

## Filing issues

The best way to get your bug fixed is to be as detailed as you can be about the problem.

Please first check the documentation at https://fluentvalidation.net first to see if your question is addressed there.

If not, then please provide the exact version of FluentValidation that you're using along with a detailed explanation of the issue and complete steps to reproduce the problem is ideal.

Please ensure all sample code is properly formatted and readable (GitHub supports [markdown](https://github.github.com/github-flavored-markdown/))

## Contributing code and content

You will need to sign a [Contributor License Agreement](https://cla.dotnetfoundation.org/) before submitting your pull request.

Make sure you can build the code. Familiarize yourself with the project workflow and our coding conventions. If you don't know what a pull request is read this article: https://help.github.com/articles/using-pull-requests.

If you wish to submit a new feature, please open an issue to discuss it with the project maintainers - don't open a pull request without first discussing whether the feature fits with the project roadmap.

Tests must be provided for all pull requests that add or change functionality.

Please ensure that you follow the existing code-style when adding new code to the project. This may seem pedantic, but it makes it much easier to review pull requests when contributed code matches the existing project style. Specifically:

- Please ensure that your editor is configured to use tabs for indentation, not spaces

- Please ensure that the project copyright notice is included in the header for all files.

- Please ensure `using` statements are inside the namespace delcaration

- Please ensure that all opening braces are on the end of line:

```csharp

// Opening braces should be on the end of the line like this

if (foo) {

}

// Not like this:

if (foo)

{

}

```

## Building the Documentation

The docs are built separately from the source code. Building the documentation requires Python 3 and pip. This is then used to install Sphinx and dependencies, which then enable `make` to build the site.

For example, on Linux / within WSL:

\* `sudo apt install python3-pip`

\* `cd docs` to go to the docs directory

\* `pip3 install -r requirements.txt` to install the required packages for the docs

\* On WSL, you may need to exit and restart at this point.

\* `PATH=$PATH:~/.local/lib/python3.8/site-packages` (you may want to add this to your `.bashrc` file as well.)

\* `make html` to build the site or `make serve` to watch for changes.