# Contributing to apaxy

First and foremost, thank you! We appreciate that you want to contribute

to apaxy, your time is valuable, and your contributions mean a lot to us.

## Important!

By contributing to this project, you:

- Agree that you have authored 100% of the content

- Agree that you have the necessary rights to the content

- Agree that you have received the necessary permissions from your

employer to make the contributions (if applicable)

- Agree that the content you contribute may be provided under the

Project license(s)

## Getting started

\*\*What does "contributing" mean?\*\*

Creating an issue is the simplest form of contributing to a project. Creating a

merge reques is more efficient form of contributing. But there are many ways to

contribute, including the following:

- Updating or correcting documentation

- Feature requests

- Bug reports

If you'd like to learn more about contributing in general, the [Guide to

Idiomatic

Contributing](https://github.com/jonschlinkert/idiomatic-contributing)

has a lot of useful information.

\*\*Showing support for apaxy\*\*

Please keep in mind that open source software is built by people like

you, who spend their free time creating things the rest the community

can use.

Don't have time to contribute? No worries, here are some other ways to

show your support for apaxy:

- star the [project](https://github.com/oupala/apaxy)

- tweet your support for apaxy

## Issues

### Before creating an issue

Please try to determine if the issue is caused by an underlying library,

and if so, create the issue there. Sometimes this is difficult to know.

We only ask that you attempt to give a reasonable attempt to find out.

Oftentimes the readme will have advice about where to go to create issues.

Try to follow these guidelines

- \*\*Investigate the issue\*\*:

- \*\*Check the readme\*\* - oftentimes you will find notes about creating

issues, and where to go depending on the type of issue

- Create the issue in the appropriate repository

### Creating an issue

Please be as descriptive as possible when creating an issue. Give us the

information we need to successfully answer your question or address your

issue by answering the following in your issue:

- \*\*version\*\*: please note the version of apaxy are you using

- \*\*extensions, plugins, helpers, etc\*\* (if applicable): please list any

extensions you're using

- \*\*error messages\*\*: please paste any error messages into the issue, or

a [gist](https://gist.github.com/)

### Closing issues

The original poster or the maintainer's of apaxy may close an issue at

any time. Typically, but not exclusively, issues are closed when:

- The issue is resolved

- The project's maintainers have determined the issue is out of scope

- An issue is clearly a duplicate of another issue, in which case the

duplicate issue will be linked

- A discussion has clearly run its course

## Pull requests

### Before creating a merge request

Please try to determine if the issue is caused by an underlying library,

and if so, create the merge request there. Sometimes this is difficult to know.

We only ask that you attempt to give a reasonable attempt to find out.

Oftentimes the readme will have advice about where to go to create merge

requests.

Try to follow these guidelines

- \*\*Investigate the issue\*\*:

- \*\*Check the readme\*\* - oftentimes you will find notes about creating

issues, and where to go depending on the type of issue

- Create the merge request in the appropriate repository

### Creating a merge request

Please be as descriptive as possible when creating a merge request. Give us the

information we need to successfully understand what you want to change or to add

to apaxy. Create the merge request as soon as possible so that the discussion

around your changes can start at the beginning of your work, not when it's over

and you don't time anymore to discuss it.

### Closing merge requests

The original poster or the maintainer's of apaxy may close a merge request at

any time. Typically, but not exclusively, issues are closed when:

- The merge request has conflict with the develop branch that makes it difficult

to merge without a hard work

- The commit message of each commit of the merge request does not follow the

[conventional commits](https://www.conventionalcommits.org/) standard

- The project's maintainers have determined the merge request is out of scope

- A merge request is clearly a duplicate of another merge request, in which case

the duplicate merge request will be linked

- A discussion has clearly run its course

## Next steps

\*\*Tips for creating idiomatic issues\*\*

Spending just a little extra time to review best practices and brush up

on your contributing skills will, at minimum, make your issue easier to

read, easier to resolve, and more likely to be found by others who have

the same or similar issue in the future. At best, it will open up doors

and potential career opportunities by helping you be at your best.

The following resources were hand-picked to help you be the most

effective contributor you can be:

- The [Guide to Idiomatic

Contributing](https://github.com/jonschlinkert/idiomatic-contributing)

is a great place for newcomers to start, but there is also information

for experienced contributors there

- Take some time to learn basic markdown. We can't stress this enough.

Don't start pasting code into GitHub issues before you've taken a moment

to review this [markdown

cheatsheet](https://gist.github.com/jonschlinkert/5854601)

- The GitHub guide to [basic

markdown](https://help.github.com/articles/markdown-basics/) is another

great markdown resource

- Learn about [GitHub Flavored

Markdown](https://help.github.com/articles/github-flavored-markdown/).

And if you want to really go above and beyond, read [mastering

markdown](https://guides.github.com/features/mastering-markdown/)

At the very least, please try to:

- Use backticks to wrap code. This ensures that it retains its

formatting and isn't modified when it's rendered by GitHub, and makes

the code more readable to others

- When applicable, use syntax highlighting by adding the correct

language name after the first "code fence"