# YARP Contribution Guide

We're excited to accept contributions, but like all open source projects need to set some guidelines on how and what to contribute. There can be nothing more frustrating than working on a change for some time, only to have it sitting forever as a PR. The purpose of this doc is to set expectations for how best to contribute so that YARP can benefit from the communities skills and knowlege.

## General feedback and discussions?

Start a discussion on the [repository issue tracker](https://github.com/microsoft/reverse-proxy/issues).

## Issues

We love issues. Issues are the best place to discuss bugs, feature requests, ideas, designs etc. Issues are the way for everyone to communicate. We use issues to communicate across the team. Almost all contributions should start with an issue.

If the conversations on an issue wander off from the initial topic, and new ideas or issues get introduced, then those should be split off into separate issues. That makes it much easier to triage the issue as the decision can/should apply to the main concept, and those separate issues won't be lost and will also be considered. So if in doubt create a new issue, and add links from both the new and original issue to each other. The new issue can always be closed if it turns out to be a duplicate.

In particular, conversations on closed issues \*won't cause an issue to be re-opened\*, and are unlikely to be noticed, so please create a new issue and refer to the old one with a link.

## Bugs

As long as humans write software, there will be bugs. If you find a bug, file an issue.

The line between a bug and a feature request or design change are tricky. For the purposes of this section a bug is where the code doesn't do what was intended by the design of the feature - it may be because of human error, or not considering cases that occur in the real world. If you feel confident about the fix, create a Pull Request (PR). Bug PR's should be small and uncontroversial, and therefore easily integrated.

## Reporting security issues and bugs

Security issues and bugs should be reported privately, via email, to the Microsoft Security Response Center (MSRC) secure@microsoft.com. You should receive a response within 24 hours. If for some reason you do not, please follow up via email to ensure we received your original message. Further information, including the MSRC PGP key, can be found in the [Security TechCenter](https://technet.microsoft.com/en-us/security/ff852094.aspx).

## Contributing code and content

We accept fixes and features! Here are some resources to help you get started on how to contribute code or new content.

\* Look at the [Contributor documentation](/docs/) to get started on building the source code on your own.

\* ["Help wanted" issues](https://github.com/microsoft/reverse-proxy/labels/help%20wanted) - these issues are up for grabs. Comment on an issue if you want to create a fix.

\* ["Good first issue" issues](https://github.com/microsoft/reverse-proxy/labels/good%20first%20issue) - we think these are a good for newcomers.

### Identifying the scale

If you would like to contribute to one of our repositories, first identify the scale of what you would like to contribute. If it is small (grammar/spelling or a bug fix) feel free to start working on a fix. If you are submitting a feature or substantial code contribution, please discuss it with the team and ensure it follows the product roadmap. You might also read these two blogs posts on contributing code: [Open Source Contribution Etiquette](http://tirania.org/blog/archive/2010/Dec-31.html) by Miguel de Icaza and [Don't "Push" Your Pull Requests](https://www.igvita.com/2011/12/19/dont-push-your-pull-requests/) by Ilya Grigorik. All code submissions will be rigorously reviewed and tested by the YARP team, and only those that meet an extremely high bar for both quality and design/roadmap appropriateness will be merged into the source.

### Roadmap

We are using project boards to track development of YARP:

\* [Planning](https://github.com/microsoft/reverse-proxy/projects/5) - this board lists the features / work items for the next major release of YARP

\* [Active Work](https://github.com/microsoft/reverse-proxy/projects/1) - this board is tracking what YARP team members are working on

Our primary focus is going to be on features and work items that are listed in the planning board for the next release. Every feature should have an issue, where scoping and design will be discussed. If you wish to contribute, we'd prefer to agree to a design in the issue, before submitting a PR. The last thing we want is for you to spend time working on a feature and then have the PR rejected or sit and get stale.

There is a cost to accepting PRs. We really appreciate your help and contributions but it takes time for us to review your code, and the team will be responsible for maintaining it. Our primary focus is going to be on work items listed on the planning board. We're happy to take a look at PRs contributing other features, but our focus will be on the work we already have planned. Those decisions aren't final though, and we change them over time as we learn new things. Feel free to file issues or comment on existing ones if you have new data to provide!

### Extensibility

One of the primary goals of YARP is to be easily extensible. Each deployment situation will be different, and the features that are "in the box" may not do exactly what you need. The goal is for you to be able to insert additional modules in the pipeline, or replace a module to achieve the functionality that you need. The answer to many feature requests may be that it should be a custom module, rather than a change to the existing feature. In those cases, we are unlikely to want a PR for the module, but will be very interested in any changes to the core that enable the extensibility for you to achieve your scenario.

### Submitting a pull request

You will need to sign a [Contributor License Agreement](https://cla.opensource.microsoft.com) when submitting your pull request. To complete the Contributor License Agreement (CLA), you will need to follow the instructions provided by the CLA bot when you send the pull request. This needs to only be done once for all repos using this CLA.

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

### Feedback

Your pull request will now go through extensive checks by the subject matter experts on our team. Please be patient; we have hundreds of pull requests across all of our repositories. Update your pull request according to feedback until it is approved by one of the YARP team members. All changes go through this process and a PR may go through multiple revisions until its accepted. This document has been through the same process, and if you look at the history there were probably multiple edits to each PR. After that, one of our team members may adjust the branch you merge into based on the expected release schedule.

## Code of conduct

See [CODE-OF-CONDUCT.md](./CODE-OF-CONDUCT.md)